Manipulation of unknown objects via contact configuration regulation

by

Orion Thomas Taylor

B.S. in Electrical and Computer Engineering, Olin College (2014)
S.M. in Electrical Engineering and Computer Science, MIT (2017)
S.M. in Mechanical Engineering, MIT (2017)

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Author . . . .
Department of Mechanical Engineering
May 22, 2023

Certified by . . . .
Alberto Rodriguez
Associate Professor
Chair/Thesis Supervisor

Accepted by . . . .
Nicolas Hadjiconstantinou
Department Graduate Officer

Manipulation of unknown objects via contact configuration regulation

by

Orion Thomas Taylor

Submitted to the Department of Mechanical Engineering
on May 22, 2023, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Mechanical Engineering

Abstract

In this thesis, we present an approach to robotic manipulation of unknown objects through the regulation of an object's *contact configuration*: the location, geometry, and mode of all contacts between the object, robot, and environment. A contact configuration constrains the forces and motions that can be applied to the object. As such, the ability to predict and regulate the contact configuration facilitates dexterous manipulation. With this as our guiding principle, we develop a joint estimation and control framework to reactively manipulate unknown objects in the gravity plane.

We begin by building a model to describe the interactions between a polygonal object, the robot, and the environment. This is accomplished by deriving the kinematic and wrench constraints associated with the geometric and frictional properties of each contact.

Our estimator generates the wrench constraints, the contact mode/geometry, and the object's shape/pose, using a combination of tactile and (limited) visual feedback. There are two separate modules: the friction estimator, which infers the friction constraints and contact mode from the measured force; and the kinematic estimator, which infers the contact geometry and the object's shape/pose from tactile and visual feedback.

The controller regulates the system's pose along the admissible motion directions, while simultaneously regulating the end-effector contact wrench to maintain the desired contact mode and geometry. The motion and wrench control objectives are balanced through a combination of a high-level controller, which synthesizes the kinematic and wrench constraints; and an impedance layer, which executes the motion.

We implement this estimation and control framework on our manipulation platform, and demonstrate that it allows the robot to reactively execute a wide variety of manipulation tasks. These include basic motions, like reorienting an object, sliding it along the ground, or performing a regrasp; as well as more advanced primitives, like using a wall as a support to reorient an object, or regulating the contact geometry between the object and the ground. Finally, we conduct ablation studies to understand the contributions from visual and tactile feedback in our manipulation framework.

Thesis Supervisor: Alberto Rodriguez
Title: Associate Professor

Acknowledgments

First, I would like to thank my advisor, Alberto Rodriguez, for his support and guidance these past nine years. He has always been very patient and kind, and I'm extremely appreciative of the creative freedom he gave me while I worked in the lab. Getting to work with Alberto has been a real treat, and he has played a key role in my development as a researcher and academic. I should also mention that Alberto welcomed both my brother, Ian, and best friend, Elliott, into the lab with open arms, and provided them with the same opportunities that he provided me. For that, I am most grateful.

I would like to thank the rest of my committee: Prof. Triantaphyllos Akylas, Prof. Neville Hogan, and Prof. Kevin Lynch. Thank you for all of the guidance and wisdom you gave me as I worked towards finishing this thesis.

It has been my pleasure to get to know so many different people during my time at the MCube lab. I thank Nima Fazeli for all of the advice he gave me during my early years. I thank Francois Hogan for teaching me how to make my own banh mi sandwich. I thank Peter Yu for being an unstoppable machine that carried us to victory in the Amazon Picking Challenge. I thank both Guillermo Fabian Diaz Lankenau and Nikhil Chavan-Dafle for our camaraderie during our years together at MIT. I thank Maria Bauza for being a shining example of hard work, perseverance, and grace. I thank Toni Bronars asking me dynamics questions: I really enjoy answering dynamics questions. I thank Rebecca Jiang for her friendship and advice, and for keeping me company in the lab during our paper crunch in January and February of 2023. Finally, I thank Elliott Donlon for our friendship these past thirteen years.

I would like to thank all of my friends and family for all of the love and support they have given me along my journey. I'm thankful for my parents, Kevin and Ruth, my siblings, Iliana and Ian, and my partner, Maya. I would also like to thank all of my teachers and mentors who, collectively, taught me almost everything that I know.

I am especially grateful to Rachel Holladay, who spent an unspeakable amount of time setting up the Panda and the computers that communicate with it. Without her work in preparing all of the necessary infrastructure, we would not have had a platform to imple-

ment the ideas presented in this thesis. I'd also like to thank Sudharshan Suresh (Suddhu) for all of his help and advice in the implementation of my factor-graph based estimator.

Last, but certainly not least, I would like to thank my good friend, collaborator, and second research advisor, Neel Doshi. The key ideas presented in this thesis were developed jointly by the two of us, 50-50. I would not have been able to get to this point without him being there for me every step of the way.

This work was supported by an ARA/Sponsored research award from Amazon.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Robots have economic value, and that value is increasing every day. Initially, robots were primarily used in manufacturing, which featured structured environments and tasks that could be broken into a sequence of simple repetitive operations: qualities which allowed for reliability during the implementation of this nascent technology. The popularity of robots has only risen since then, and nowadays they can be found everywhere: in warehouses, transporting people to work, and operating in environments that are too dangerous for humans (just to list a few examples). One day they may even care for our elderly, fix our plumbing, or cook complex recipes in the kitchen.

However, despite the incredible advances of machine learning and artificial intelligence (e.g., AlphaGo, GPT-4, and Stockfish), humans still have robots completely beat when it comes to manipulation. Folding a paper airplane, tying a shoelace, or putting on a sweater are all tasks that are easy for a human (even without the use of sight) but quite difficult for a robot; and not a robot exists today that could do all three. If we don't want to get accidentally crushed to death by our robot caretakers when we're older, then we need to bridge the gap in dexterity between robots and humans.

To make robots more dexterous, we need to understand contact. Contact is the interface through which robots are able to interact with their surroundings. As such, the mechanics of contact govern what a robot can and cannot do. In this thesis, we use existing models of contact mechanics to develop a joint estimation and control framework that facilitates reactive and dexterous interactions between a robot and the environment.

1.1 Contact Configuration Regulation

Regulation of an object's *contact configuration* – the location, mode and geometry of all contacts between the object, robot, and environment – is a fundamental abstraction of object manipulation (Figure 1-1). Imagine tumbling a heavy box or tightening a screw. Both tasks can be better described/executed by prescribing/regulating the location, geometry, and mode of all contacts. In these cases, the object can be sufficiently controlled without using pose, inertial, or shape information. Even when this information is available, contact configuration regulation simplifies control, for example during non-prehensile [6, 25] or deformable-object [61] manipulation. As such, contact configuration regulation can be used to manipulate unknown objects (Figure 1-1b). This is a *joint estimation and control* problem. The robot must estimate the kinematic and frictional constraints imposed by the contact configuration and regulate the contact forces and object motion accordingly. This is challenging, as not all contacts are directly observable, and the robot's control authority is limited by the underactuated mechanics of friction. We focus on manipulating unknown planar objects on a flat surface using robot proprioception and force/torque sensing at the wrist for feedback (Chapter 2). This minimal system has a diverse set of contact configurations that highlight the challenges discussed above.

1.1.1 Vocabulary

Before we proceed, it would be best to take a moment to briefly define a few terms that will appear frequently throughout this work. Over the course of this thesis, we will explore each of these concepts thoroughly (especially in Section 2).

- **Contact geometry:** In this work, when we use the term contact geometry, we are specifically referring to the pair of geometric elements that are interacting to form a contact. We consider two types of contact geometries: point/line and line/line (also referred to as flush contact), which are illustrated in Figure 2-8.

- **Point/line contact:** During point/line contact, a corner of one planar body is in contact with the face of another planar body (Figure 2-8 left).

- **Flush contact:** During flush contact, the faces of two different planar bodies are in contact with one another (Figure 2-8 middle).

- **Contact mode:** In this work, when we use the term contact mode, we are specifically referring to whether or not a contact is sticking or slipping. The contact mode is determined by the contact forces and the friction properties of the contact.

- **Contact configuration:** A system's contact configuration consists of the location, mode, and geometry of all contacts between the object, robot, and environment.

- **Admissible motion:** An admissible motion is a motion of the system that is consistent with the active kinematic constraints.

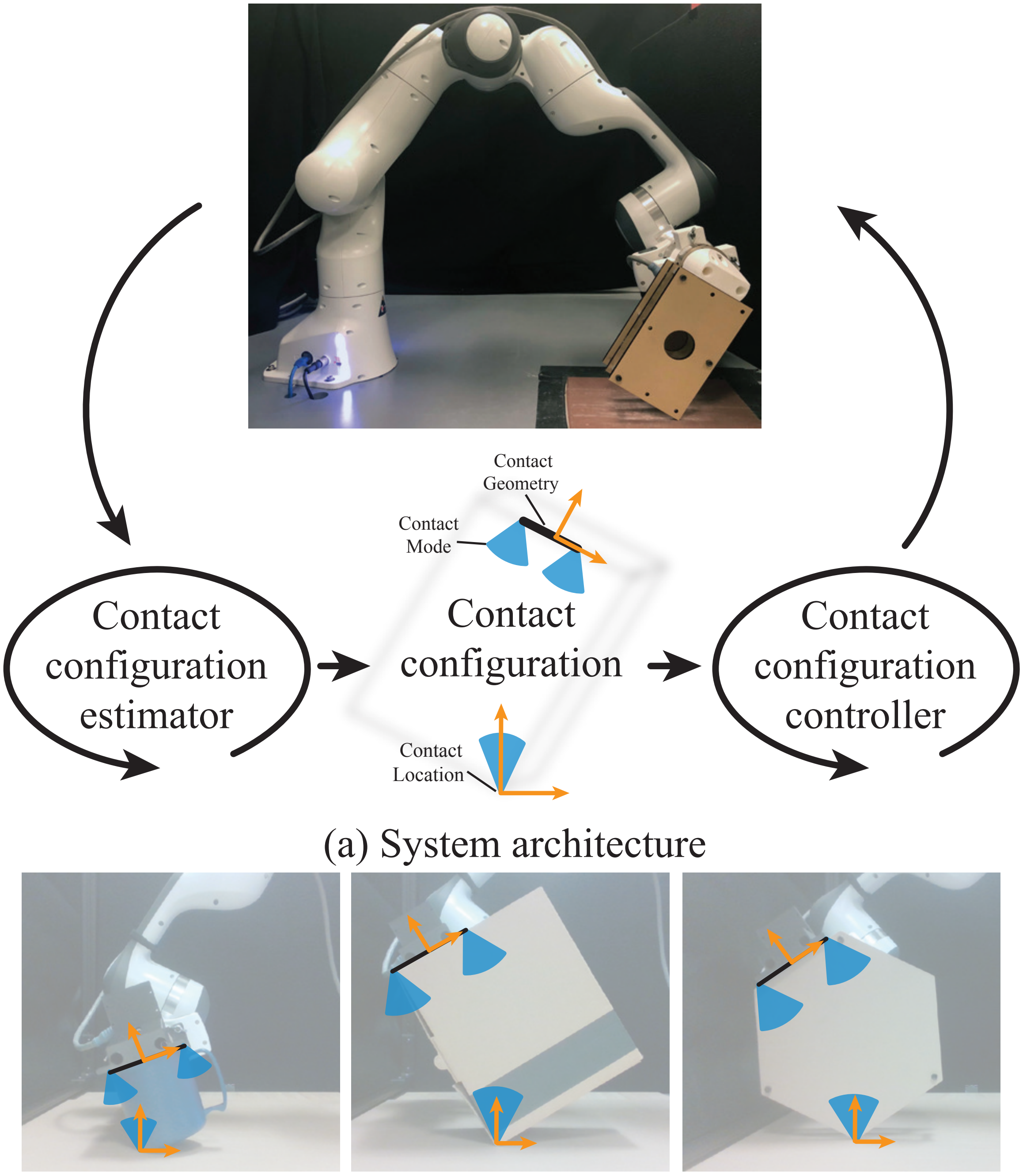
- **Admissible motion direction:** An admissible motion direction is a vector that is parallel to an admissible motion of the system.

1.2 Related Work

There is a significant amount of previous work that overlaps with ours in some way or another. We have identified a few key themes that our work shares with other lines of research: hybrid dynamics, extrinsic contact, system identification of contact mechanics, object shape and pose estimation, simultaneous force and motion regulation, joint estimation and control, and contact configuration regulation. We will now take a look at some of the work that is adjacent to our own.

1.2.1 Hybrid Dynamics

Hybrid dynamics are characterized as having both continuous and discrete behavior. Any system with contact mechanics (which includes many robotic manipulation and legged locomotion systems) has hybrid dynamics of some kind, since the positions and orientations of the various system elements (as well as the forces and torques) are continuous variables, while the contact states (e.g., sticking/slipping or contact/no-contact) are discrete.


Contact
Geometry
Contact
Mode
Contact configuration estimator
Contact configuration
Contact configuration controller
Contact
Location

(a) System architecture

(b) Contact-configuration based manipulation

Figure 1-1: (a) The robot interacts with an object by estimating and controlling its contact configuration. (b) This enables manipulation of unknown objects without explicit knowledge of their geometric and inertial properties.

This juxtaposition of the continuous and discrete presents a challenge for modeling, simulation, planning, estimation, and control because many of the standard mathematical tools for solving these problems rely on the assumption that a system is either entirely continuous or entirely discrete. For example, continuous algorithms, like gradient descent and Newton's method, do not handle discontinuities very well and are thus likely to fail when naively presented with a hybrid system which, as a result of their discrete behavior, always contains discontinuities of some kind. Similarly, discrete algorithms, like the many different flavors of graph search, often perform poorly on hybrid systems that have been discretized, usually because of the curse of dimensionality. That being said, when designing a planner or controller for an underactuated robotic system, it is often possible to exploit the hybrid dynamics of the system in order to access a larger region of the state space. For example, a legged robot might be able to jump over an obstacle that it couldn't directly walk through.

Given both the hybrid nature of contact mechanics, and how fundamental contact mechanics are to manipulation and legged locomotion, it should come as no surprise that a significant amount of robotics research deals with hybrid dynamics in some way or another. We begin with Lynch and Mason [44, 45], who build a motion planner for a robot consisting of a single paddle mounted to a motor. Despite only having a single degree of freedom, their robot is able to exploit the hybrid nature of frictional contact to execute a diverse set of manipulation primitives on an object, including snatching, throwing, or rolling. When properly sequenced, these primitives can be used to drive an object from between an arbitrary pair of initial and target states, though in this case the schedule of modes is manually provided. This work is built on further by Woodruff and Lynch [69], who extend their framework to a robot with several degrees of freedom, develop several closed-loop controllers to stabilize the various motion primitives, and build a planner to compute the correct sequence of motion primitives to drive the object to the target state.

Hogan and Rodriguez [25] design a reactive controller for the problem of planar pushing. During planar pushing, a robot pushes an object that is resting on a table, with motions generally being limited to horizontal translations and rotations in the plane of the table surface (this is a result of the problem design: there aren't actually any kinematic constraints in place to prevent other motions from occurring). In the case of [25], the robot end-effector (idealized as a single point or a pair of points) pushes the object (idealized as a closed contour in the plane) at its boundary, with the hybrid dynamics arising from the possibility of either sticking or slipping between the object and end-effector. Using a combination of model predictive control (MPC) and a mode scheduler, the controller they develop is able to drive the object along a desired path while rejecting any perturbations to the system. This work is continued by Hogan et al. [26], who design a closed-loop controller and a contact state planner for a dual-arm robot to perform a more sophisticated version of planar pushing (in this case the object can tumble, allowing for out of plane rotations with respect to the table surface).

Cheng et al. [7] present a sampling-based planning framework that plans motions and contact mode transitions simultaneously, allowing it find and execute trajectories that cannot be easily described as a sequence of macro motion primitives. Philosophically, this is quite different from our approach to hybridness. Specifically, our controller executes individual motion primitives (which specify a discrete contact state and a pose displacement), with the goal of reducing the planning problem to that of scheduling these macro primitives.

Hybrid systems can also present a significant challenge when it comes to estimation. In the context of our problem, we must use tactile feedback to disambiguate between different potential contact geometries between the object and environment. Determining how to process a tactile measurement requires knowledge of the contact geometry, but correctly estimating the contact geometry requires that the tactile measurement was processed in the first place. This is called the *global assignment problem* (or data association problem), and it shows up all the time in estimation. Our approach to this is rather blunt: we use a combination of a prior generated by vision and a set of heuristics to disambiguate discrete contact states. There are more sophisticated approaches however, like multi-hypothesis models [32, 13] or particle filters [54].

1.2.2 Extrinsic Contact

In the field of robotic manipulation, the term extrinsic contact is used to refer to contact interfaces between the object and the environment. Though they significantly affect the dynamics of the system, the geometry (shape of the contact interface), frictional properties (friction coefficient or limit surface) and state (pose and contact wrenches) of these contacts usually aren't measured directly, and must be inferred from measurements taken further up the kinematic chain. It is possible to exploit the wrenches generated by environmental contacts in order to add extra degrees of actuation to the system. Examples of this include "graspless manipulation" [1], using surfaces in the environment to push an object into a gripper [16], or using an extrinsic contact to support a pivoting in-hand regrasp [28].

Ma et al. [47] build an extrinsic contact estimation framework for a grasped object. In addition to robot proprioception (which measures the pose of the robot end-effector) and a force/torque sensor in the wrist of the robot, their system is equipped with a pair of GelSight sensors in the fingertips of the robot. GelSight is a high resolution tactile sensor that is able to measure the displacement field of a contact patch at the fingertip. By combining these measurements with the kinematic constraints of sticking contact, they are able to estimate the position and geometry on an environmental contact. Kim and Rodriguez [38] build on this framework, incorporating a feedback controller to enforce the desired contact mode (e.g. sticking or slipping) at the extrinsic contact, which facilitates estimation. It should be noted that their estimation framework makes use of factor graphs, a popular estimation tool that we also employ.

Chavan-Dafle et al. [6] design a planner that performs in-hand regrasps of an object by pushing it against features in the environment (walls, the ground, etc.). This manipulation system is a sort of dual to planar pushing, where the roles of end-effector and environment are reversed. Specifically, the environment is used to actuate the object (relative to the gripper), and the parallel jaw grasp of the object forms the work surface that the object travels along. Their manipulation framework explicitly reasons about the set of feasible motions that can be imparted on the system (the motion cone) in order to plan the sequence of pushes that moves the object towards the target pose. We use similar reasoning (admissible motion directions) to design a controller that can regulate our system along the available degrees of freedom.

1.2.3 System Identification of Contact Mechanics

A core part of our estimation framework involves inferring the contact wrench constraints. Specifically, we estimate the generalized friction cone [3, 17], a structure that synthesizes both the friction constraints (which determine whether or not a contact is sticking of sliding) and the torque constraints (which are associated with the geometry of the contact interface). With the estimated generalized friction cone, it is possible to both predict and regulate the contact mode (sticking/sliding) [53, 36] and geometry (point/line vs. flush contact).

Prior work in this field has focused on the problem of planar pushing [70, 46, 43, 72]. Here, the goal is to use measurements of both the wrenches exerted on the object and the resulting motions in order to estimate the input-output mapping between the two. The type of underlying parameter being estimated varies from work to work. Lynch et al. [43] and Yoshikawa et al. [70] focus on estimating the center of friction, which is kind of like the quasi-static analog to the center of mass, in that it determines whether or not an object will rotate clockwise or counterclockwise depending on the relative location of the applied force [48, 51]. On the other hand, Zhou et al. [72] directly estimate the limit-surface [20], a structure related to the generalized friction cone, that describes the set of wrenches that can be transmitted through planar frictional contact. We find this work to be particularly relevant because it tries to directly estimate the relationship between wrenches and velocities as opposed to estimating the underlying frictional and geometric parameters. In a similar fashion, our estimator directly estimates the frictional wrench constraints instead of the underlying parameters (the friction coefficient and object weight). Taking this idea even further, estimation of the friction constraints can be eschewed in favor of directly learning a model for sticking and slipping [66].

1.2.4 Object Shape and Pose Estimation

Our framework synthesizes visual and tactile information to estimate the location, geometry, and mode of all contacts between the robot, object, and environment. There is a large body of work in the area of localizing contacts/objects, estimating contact configurations, and reconstructing an object's shape, using vision and/or tactile feedback. Many of these works present solutions to specific aspects of the problems we tackle in this thesis. Prior information about the end-effector geometry can be exploited to localize contacts [4, 49, 71, 67] or estimate a stationary object's shape [34]. Similarly, knowledge of the object geometry can be used estimate its pose using only tactile sensing [57], or a fusion of both visual and tactile feedback [5, 23, 33].

Particle filters are a popular formulation for nonlinear estimation problems, and have been often used in the context of tactile localization [60, 39, 41, 52, 63]. Koval et al. [39] develop an approach to particle filters that takes into account how contact restricts an object's degrees-of-freedom. Li et al. [41] explicitly reason about the complementarity constraints of frictional contact, while Meeussen et al. [52] apply particle filters to fuse visual and tactile data. Sipos and Fazeli use particle filters to estimate both the object pose and the location of extrinsic contacts [63]. These works partially overlap with the problem that we are trying to solve.

Several approaches have been presented to the problem of joint shape and pose estimation for 2D objects [64, 65, 71]. The estimation framework is called tactile SLAM (simultaneous location and mapping). Strub et al. [64] focus on the problem of estimating the shape of a regular polygon that is constrained to rotate about its centroid, while it is being manipulated by a two-fingered robot. Yu and Rodriguez [71] and Suresh et al. [65] focus on the problem of estimating a variety of 2D shapes during planar pushing. Both of these works make use of factor graphs [35, 11] (a popular estimation tool in the SLAM community) to encode the kinematic constraints of contact, a method that we also employ. This technique of using factor graphs to represent contact constraints is not unique to the manipulation community: for instance, Hartley et al. [22, 21] use contact factors to estimate the position of a legged robot. Unlike these works, we place a significant emphasis on estimating extrinsic contact formations that are not measured directly.

1.2.5 Simultaneous Force and Motion Regulation

Our control framework regulates both the discrete contact state of the target object (sticking/slipping and contact geometry) and its pose. Since these are respectively determined by the contact wrenches and end-effector velocities, our framework must be able to reason about both forces and motions simultaneously. To accomplish this, the controller solves a quadratic program (QP) at every time-step. The cost and constraints of this QP correspond to the motion and force regulation objectives respectively. The solution of the QP is a target pose, which is then handed to a low-level impedance controller (a controller that emulates the behavior of a mass-spring-damper system) to execute.

This approach to simultaneous force and motion regulation can be viewed through the lenses of two different control frameworks: indirect force control (IFC) [59, 27] and hybrid force-velocity control (HFVC) [50, 58, 30] (and perhaps also its sibling, hybrid position-force control [68]). IFC regulates wrenches and motions by prescribing the interaction dynamics, while HFVC directly regulates wrenches and motions in orthogonal subspaces. From the IFC perspective, our framework is similar to hybrid impedance control [2], where wrenches and velocities are controlled in orthogonal subspaces via impedance modulation. From the HFVC perspective, it's similar to parallel force/velocity control [8], where separate wrench and velocity controllers are summed, with priority given to the wrench controller. However, our approach is unique in that, instead of defining orthogonal subspaces or fixed priorities, we use the aforementioned quadratic program to determine wrench and velocity control directions and priorities online. In the sense that our contact configuration controller regulates a pose as a response to force feedback, which is then passed to an impedance controller, our framework could be perhaps thought of as a series admittance-impedance controller [18].

1.2.6 Joint Estimation and Control

We present both an estimator and a controller in this thesis. The two were built at the same time, and are designed to work together. The estimator relies on the controller to excite the object in stable manner, so that it can infer the kinematic and wrench constraints of the system. The controller uses these constraints to determine which directions it can move the object, and how to regulate the wrench in order to enforce the desired contact state.

While there is more recent work on joint estimation and control, individual papers often make simplifying assumptions (e.g., frictionless models of contact) [40, 10], or learn task-specific policies from data (e.g, for cable manipulation [61], part insertion [14], or manipulating rigid objects on a shelf [42]). In contrast, our framework is object-agnostic and reasons about all frictional interactions between the robot, object, and environment, leveraging limited visual feedback to execute dexterous manipulation tasks.

1.2.7 Contact Configuration Regulation

The contact configuration of an object is the location, mode, and geometry of all contact formations between it and the robot/environment. This is a useful abstraction in robotic manipulation, because contact is required to impart a wrench/motion on the object. The idea is that regulating the behavior at the contact interfaces of a system is often sufficient for object manipulation, and in the cases where it isn't, contact configuration regulation can still simplify estimation and control.

Prior work on contact configuration regulation has focused on tasks such as polishing [19], deburring [24], opening a variety of drawers [37] or doors [55], or manipulating a triangle with two fingers [36], cases for which the contact interface being regulated is measured directly (facilitating estimation and control).

Other works (some of which we have already discussed) have focused on systems where the behavior of an extrinsic contact is relevant. In this area, prior research usually either focuses on control assuming a known model of the world [31, 26], or estimation assuming stable interactions [47, 62]. We have been particularly inspired by the work done by Hou et al. [29, 30, 31], who use hybrid force velocity control (HFVC) to partition discrete contact state regulation and pose regulation into two separate problems. This approach is directly adjacent (and extremely relevant) to our own. It is worth looking at the differences between the two frameworks, because they highlight several of our key research contributions:

- Hou et al. explicitly separate the degrees of freedom of the system into orthogonal subspaces for force and velocity control (for the respective purposes of enforcing sticking contact and moving the object). On the other hand, we use the combination of a quadratic program and an impedance layer to balance between those two control priorities online, an approach that is naturally more forgiving of estimation/modeling error.

- For their framework, robustness comes from planning an open-loop trajectory of poses and wrenches that maximizes a stability margin for both the trajectory (geometrical stability margin) and the compliance directions along the trajectory (control stability margin). In contrast, the robustness of our controller originates from its ability to react to pose and wrench feedback in a closed-loop fashion.

- Finally, we have designed an estimator to infer the geometric and frictional properties of the object and environment, instead of relying on a nominal model of the system.

1.3 Contributions

The main contributions of this thesis are a framework for regulating contact interactions, and a series of system choices that makes this possible.

Friction parameter estimation: The behavior of a manipulation system is governed to first order by the friction coefficients at its contacts. A conservative assumption of the friction coefficients reduces the effective control authority of the robot, since the set of forces and motions it thinks it can impart on the object have been reduced. On the other hand, if the controller has been provided a nominal value of the friction coefficient that is too high, it may unintentionally exert forces that result in sliding contact when sticking was commanded.

As such, the ability to dynamically estimate the frictional properties of a contact is useful for manipulation. Our approach to this problem is to directly estimate the set of forces that can be transmitted through a contact, as opposed to estimating the underlying system parameters (like the friction coefficient or weight of the object). This estimate is then handed to the controller for the purpose of contact wrench regulation.

Simultaneous object shape and pose estimation: We do not have access to ground truth on the object's pose and geometry. The estimator must infer both from the motion and wrench measurements of the end-effector. This is aided to some extent by a prior on the object's shape, which is used to seed the estimation loop with an initial guess.

This problem of simultaneously estimating both the geometry and pose of a system is called SLAM (simultaneous location and mapping), and is often seen in the context of mobile robots trying to map their surrounding while determining their current position and orientation in the map that they've constructed. The manipulation variant of this problem is called tactile SLAM, and involves estimating the pose and geometry of a manipulated object using some combination of tactile sensing, robot proprioception, and vision.

Extrinsic contact estimation/regulation: In our system, the contact interface between the object and environment plays an outsized role in the system mechanics, and yet its behavior cannot be measured or controlled directly (unlike the hand/object contact). From both an estimation and controls perspective, both the observability and controllability at this contact is limited by the fact that the sensing and actuation occurs through the hand/object interface. The framework we have designed is able to deal with this however, and in many cases we are able to exploit the extrinsic contact to help actuate the system.

A modular approach to hybrid manipulation: As we have discussed, estimating and controlling hybrid systems can be very challenging, and the hybridness of contact is at the very core of our manipulation problem. Our approach to this is to enumerate the kinematic and wrench constraints associated with each type of contact interaction (e.g., stick vs. slip at the hand contact or point/line vs. flush contact at the ground contact). These are the building blocks that are used by the estimator and controller, with each discrete contact state corresponding to a composition of building blocks. Once the discrete contact state has been inferred (estimator) or commanded (controller), the estimator and controller assemble the individual blocks into a math problem whose solution is either the system state and parameters (estimator) or a control action (controller). In the case of the estimator, this math problem comes in the form of a factor graph, where each of the constraint factors correspond to a kinematic or wrench constraint associated with the inferred discrete contact state. For the controller, the math problem is a quadratic program, where the costs and constraints respectively correspond to the admissible motions and wrench constraints of the commanded discrete contact state.

Partitioning the planning and control problems for a hybrid manipulation system: We have developed a controller that is able to execute a variety of motion primitives. Each motion primitive corresponds to the combination of a commanded discrete contact state and a desired pose displacement of the system. The controller then acts to enforce the contact state while driving the system along the displacement direction.

Currently, the sequence of motion primitives is chosen by a human, but they could conceivably be chosen by a planner. Ideally, given an initial and desired pose of the system (that are not necessarily reachable via a single motion primitive), the planner would string together a sequence of primitives to drive the system to its target state. In this sense, in designing our controller, we have partitioned planning and control into two separate problems. First, the planner would generate a schedule of primitives, reasoning about which motions are feasible from each intermediate state. Then, the controller would execute each motion primitive in the sequence. This approach stands in contrast to some hybrid trajectory planners, which have to reason about different discrete modes, the transitions between the modes, and the kinematic/wrench constraints associated with the modes, for each step along the trajectory. The computational complexity of such a problem can grow pretty quickly, which can be quite limiting.

Reactive tactile control: Our control system is closed-loop w/respect to the object/hand poses and the contact wrenches of the system. As such, it is able to dynamically respond to external perturbations. Additionally, the combination of the quadratic program (QP) and impedance layer provide a built in level compliance that is forgiving of estimation error.

Codesign of estimation and control systems: In our framework, the estimator and controller were both designed with the other in mind, and have equal importance. The wrench constraints and admissible motion directions inferred by the estimator are used by the controller to regulate the impedance target. The estimator relies on the stable contact interactions generated by the controller in order to excite the system to collect useful data. Our controller does not rely on a nominal model of the system (or ground truth), since we estimate both the current state (poses and wrenches) of the system, as well as the underlying frictional and geometric parameters of the object and environment. As such, our framework is object-agnostic (it can handle a variety of object shapes, which do not need to be specified as input beforehand). This is a fully integrated manipulation system.

Contact configuration regulation as a unifying manipulation framework: In contact configuration regulation, emphasis is placed on estimation/control of the properties/behavior of the contact interfaces of a system. Everything else is secondary to this. For instance, though our estimator does build a model of the global geometry and pose of the object, this is in service of estimating the location and geometry of the hand and ground contacts. Similarly, control primitives are expressed in terms of motions of the contact locations and contact wrench constraints. We avoid directly dealing with the global properties of the system (like the object's mass), preferring to focus on local contact behavior instead.

As previously stated, the idea here is that reasoning about the contact behavior of a system is often sufficient for object manipulation, and in the cases where it isn't, doing so can still simplify estimation and control. The contact configuration of a system (the location, mode, and geometry of all contact formations) is a fundamental abstraction to manipulation that should be exploited.

1.4 Outline

This thesis is a synthesis of two works, one of which has already been published ([15]), and one of which is currently under review. Though some of the content in this thesis has been taken verbatim from these two papers, a significant amount of extra detail has been included here that previously had to be omitted due to page limitations.

We begin by presenting a model for our system in Chapter 2. This is the longest chapter. We really try to give the model a thorough treatment, because it is the foundation on which both the estimator and controller are built. This is split into three parts: a bird's eye view of our research platform (Section 2.1), an introduction to the underlying physical/mathematical principles that govern the behavior of our system (Section 2.2), and finally a derivation of the kinematic and wrench constraints specific to our system (Section 2.3).

From this model, we derive our estimation framework in Chapter 3. This estimator consists of two modules: the friction estimator and the kinematic estimator. The friction estimator (Section 3.1) infers the friction constraints and the stick/slip state at the contacts from the measured contact wrenches. This process involves approximating the 2D convex hull for a set of measured forces, a process which we describe in Appendix A. The kinematic estimator, described in Section 3.2, infers both the contact geometry and the shape and pose of the object, using both tactile feedback and a limited amount of visual feedback. Our vision system uses a heuristic that we developed (described in Appendix B) to extract the vertex positions of a polygonal object from an image.

The estimates of the wrench constraints and the object's shape/pose are then passed to the contact configuration controller, which we describe in Chapter 4. This controller is built on top of a low-level impedance controller (Section 4.1) which emulates the behavior of a mass-spring-damper system. The contact configuration controller itself (Section 4.2) solves a quadratic program at every time-step in order to find the pose of the impedance target that will regulate both the contact wrench and the object pose in the desired way. The composition of constraints and cost terms in this quadratic program is determined by a motion primitive command (Section 4.3), which specifies a desired discrete contact state and pose displacement of the system.

We perform several experiments and demonstrations in Chapter 5 to test the performance of our system and show our framework in action. Finally, we conclude with a short discussion in Chapter 6 of potential directions for future work, and the lessons we learned along the way.

1.5 About the Author

Greetings. Thank you for taking a look at this thesis. My name is Orion Taylor. I'm currently finishing up my ninth (and last) year of graduate school at MIT. I really enjoy math, engineering, anime, manga, western animation, Magic the Gathering, Pokemon, piano, video games, sailing, fishing, the beach, competitive coding, spending time with my friends, and robotics. I went to Lake Montessori for elementary school, Trinity Prep for middle and high school, and the Olin College of Engineering for my undergraduate degree. I started at MIT the semester after I graduated from Olin, and have been there ever since. I am the last remaining student of the MCube lab from the initial batch that joined when the lab first formed back in the fall of 2014.

This thesis only covers the last three years of research that I did in graduate school. During my first six years, I worked on the codesign of shapes and trajectories of contact juggling robots. As part of the lab, I also participated in the Amazon Robotics (Picking) Challenge for three years. This was a competition where the goal was to design a robotic system to pick items out of a box and place them into a warehouse shelf. My main role was to design the individual motion primitives that the robot would execute.

In addition to this, I spent eight consecutive semesters (four years) as a teaching assistant for dynamics at both the graduate and undergraduate levels (2.032 and 2.003 respectively). I believe this experience had a significant impact on the work in this thesis.

On the recreational side of things, I started taking piano lessons four years ago (after having quit when I was a child). I also played more World of Warcraft Classic than I would care to admit during the year of the pandemic (I was my guild's heal team leader, and they gave me an Atiesh). I have been a Magic the Gathering fanatic from the moment that my friend Elliott Donlon taught me how to play when I was in college. Finally, I really enjoy coding puzzles. I've been doing a fair amount of LeetCode and Advent of Code, and I plan to start doing more difficult challenges once I'm done with my thesis.

I love being a student. I've been a student for my entire lifetime. I wish I could stay even longer, but all good things must come to an end.

Chapter 2

Modeling

Our manipulation framework is entirely model based. To understand how the estimator and controller work, we must first derive the kinematics and dynamics of our system. We begin this section by providing a general overview of our manipulation platform, defining relevant terms, and describing our modeling assumptions. We then explore the underlying physics that governs the behavior of the system. Finally, we derive the kinematic equations and wrench constraints for our system, which will be used in the development of our estimator (Chapter 3) and controller (Chapter 4).

2.1 System Description

We focus on quasi-static manipulation of unknown planar objects on top of a horizontal work surface. Motion is constrained to the sagittal (vertical) plane, meaning that gravity acts within the plane. As such, there are configurations of the system for which, if the robot hand leaves contact with the object, the object will topple over. In these configurations the system is, in some senses, unstable. This is a bit different from the canonical problem of planar pushing: since gravity is often normal to the motion plane in planar pushing, the object never needs the support of the robot hand to remain at rest.

As depicted in Figure 2-1 (top left), the system consists of three main components:

- **The robot hand** (or palm) which is essentially just a flat surface (and is modeled as a line segment in the plane of motion). The hand is attached to a robot arm, which we will treat as a black box that simply serves to actuate the hand (Section 4.1).

- **The target object**, which is a prism whose front and back faces are convex polygons (and is modeled as a polygon in the plane of motion). In our experimental setup, we construct this polygonal prism by bolting together three identical laser cut shapes. Why three shapes when two would suffice? We want our prism to have the same length (in the direction normal to the plane of motion) as the robot hand, in order to keep it from either toppling over or twisting out of the desired plane of motion while it is being manipulated. We are cannibalizing metal standoffs from a previous research project, which are half the necessary length, so we needed to stack three shapes together instead of two to get the desired length.

- **The environment**, which consists of the ground and up to two vertical walls, which are modeled as horizontal/vertical lines in the plane respectively.

The system has access to three forms of feedback:

- **Robot Proprioception**: The robot arm control interface provides direct access to the pose (position and orientation) of the robot end-effector (hand/palm), also known as the tool center point (TCP). The pose is computed by applying forward kinematics to the joint angles of the robot arm (which are measured via encoders in the joints of the robot) and the linkage lengths of the arm (which are known by the manufacturer).

- **Force/Torque (F/T) Sensing**: We use a Force/Torque sensor in the wrist of the robot (see Figure 2-1 top right) to measure the contact wrench (force + torque) that the robot exerts on the object. We are using an ATI Gamma, calibrated for a maximum force/torque of 32 N and 2.5 Nm respectively. This sensor has a resolution (in the force/torque directions we care about) of 1/160 N and 1/2000 Nm.

- **Vision**: A camera mounted to the robot arena (see Figure 2-1 top left) gives us access to visual feedback of the system. Unlike tactile sensing (proprioception and F/T sensing), we usually do not make use of live visual feedback. Instead, vision is often used to generate an initial estimate of the object pose and geometry, which is used to warm-start the main estimator.

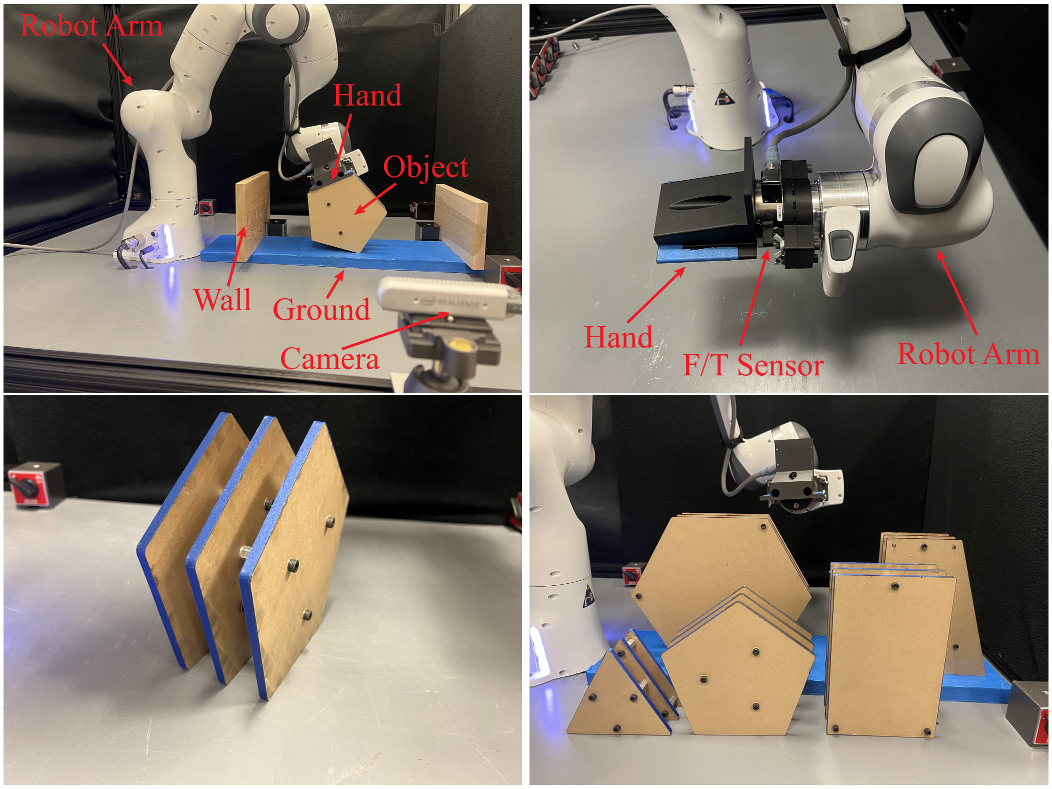


Figure 2-1: Images of the experimental setup. **Top left**: the various components of our planar manipulation system. Note that the ground is elevated using a block of wood. This change in ground height must be inferred by the estimator. **Top right**: we use a force/torque sensor in the wrist of the robot to measure the contact wrench exerted by the robot on the object. **Bottom left**: though we model the system as being planar, the experimental implementation exists in the real world, which is 3D. To approximate planar behavior, our object consists of three identical laser cut shapes that have been bolted together. **Bottom right**: we primarily focus on manipulating convex polygons.

Through our simplifying assumptions we have constructed a toy problem that distills a larger class of dexterous manipulation tasks (part insertion, tool use, stacking, assembly etc.) down to a few shared essential features:

- **Contact-rich Manipulation**: The dynamics of this system are governed by contact mechanics. The object's motion is determined by the wrenches and motions that the robot and environment impart on it through the contact interfaces of the system.
- **Underactuation**: The robot cannot move the object in any direction arbitrarily. Its control authority is limited by both friction and the geometry of each of the contacts.
- **Hybrid Dynamics**: This system has both continuous and discrete behavior. Specif-

ically, the object pose (position + orientation) and the contact wrenches (forces + torques) are continuous variables while the mode (sticking/sliding) and geometry (point contact/line contact/no contact) of each of the contacts are discrete.

- **Extrinsic Contact**: The contacts between the object and the environment (the extrinsic contacts) play a significant role in the system dynamics. Since we do not equip the environment with tactile sensors, the behavior/properties of the extrinsic contacts cannot be measured directly, and must instead be inferred from measurements taken at the robot/object contact interface. Even though they present a challenge for estimation, extrinsic contacts can be used as a resource during manipulation [9].

- **Diverse Set of Behaviors**: Even though this system is somewhat simple, the set of feasible motions is quite diverse. For instance, as depicted in Figure 2-2, it is possible to reorient the object by pivoting it about an external contact, translate the object by sliding it along the ground, or perform a regrasp by sliding the hand relative to the object. These are just a few examples of the possible types of state transitions.

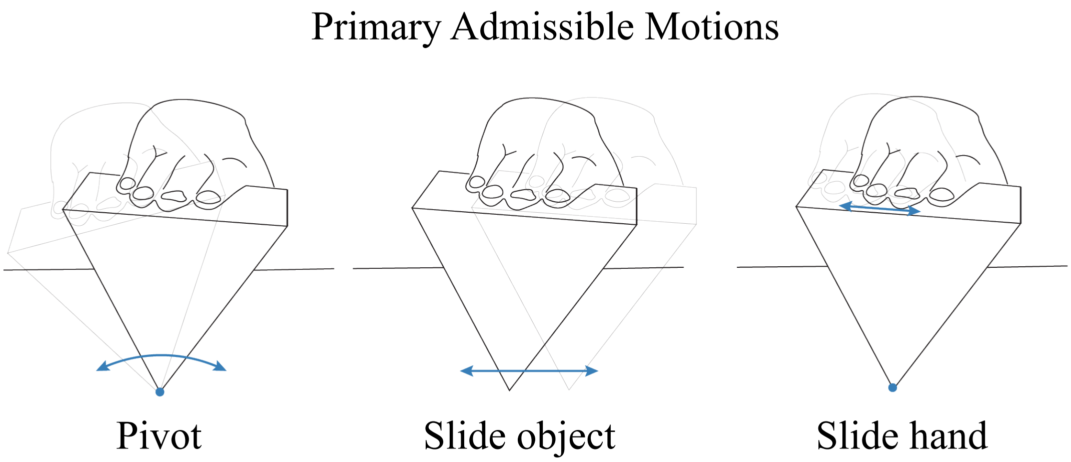


Figure 2-2: The primary set of admissible of motions of the system. **Left**: reorienting the object by pivoting it about an external contact. **Center**: translating the object by sliding it along the ground. **Right**: performing a regrasp by sliding the hand relative to the object.

2.1.1 Modeling Assumptions

We make several simplifying assumptions to make modeling the system easier. First and foremost, we assume that the system is quasi-static: in other words, the system is in static equilibrium at every instant in time. As such, the net wrench on the object is zero:

$$\sum \vec{F} = 0, \quad \sum \tau = 0 \tag{2.1}$$

We rely heavily on (2.1) when developing our model of the system. The quasi-static assumption is a good approximation when the system's inertia is relatively small compared to the damping. This results in the kinetic energy decaying quickly, meaning that the system returns to rest soon after being perturbed. For our system, the primary sources of energy dissipation are friction during sliding contact, and (more importantly) the virtual damping in the low-level impedance controller of the robot arm. The objects we manipulate are small (around 20 cm in diameter) and light (.2kg-1kg), so both the mass and moment of inertia are small compared to the translational/rotational damping in the impedance controller (which are set around 22 N-s/m and 4 N-m-s respectively). This corresponds to a decay time constant somewhere on the order of magnitude of .01-.1 seconds (computed using the formula $\frac{m}{b}, \frac{I}{b_\theta}$ for a mass-damper system or $\frac{2m}{b}, \frac{2I}{b_\theta}$ for an underdamped mass-spring-damper system, where we are not taking into account the inertia of the arm).

We use a planar model to represent the system. All motion is constrained to the sagittal (vertical) plane. This means we can model elements in our system as 2D shapes, with each individual element having at most three degrees of freedom (vertical/horizontal translation and in-plane rotation). A 3D system can be modeled as planar if its behavior is isotropic (uniform) in the direction normal to the plane. In the case of our experimental implementation (which is 3D), we approximate this by making the target objects prisms, with their corresponding 2D model being the shape of their front/back face. In addition, the stiffness parameters of the low-level impedance controller are chosen to minimize out-of-plane rotation/translation of the robot hand.

We model the hand/object/environment as rigid polygons/line-segments. This simplifies their parameterization, allowing us to represent them using their vertices and surface normals. As a consequence of the rigid-body assumption, the hand/object/environment do not deform, which means that we can keep track of their boundaries over time using only their poses. This is approximately true for our system because, for our choice of materials, we do not load the system elements enough to generate significant deformations. Finally, we assume that the ground and walls are horizontal and vertical respectively.

2.1.2 Contact Geometries

In this work, when we use the term **contact geometry**, we are specifically referring to the pair of geometric elements that are interacting to form a contact. Since we are using a planar model to represent our system, surfaces and edges in 3D are represented as lines and points respectively (which are the two types of geometric elements that can interact). Thus, the three possible types of contact geometries are point/point, point/line and line/line (which we refer to as flush contact). However, we usually ignore the possibility of point/point contact because it rarely occurs in our system and is inherently unstable. Each contact geometry comes with its own set of kinematic and wrench constraints, which we will discuss in depth later. The different types of contact geometries between the object and the hand/environment are enumerated in Figure 2-3.

2.1.3 System Parameterization

The system's state consists of the planar poses of the hand $\vec{x}_h = (\vec{r}_h, \theta_h)$ and object $\vec{x}_o = (\vec{r}_o, \theta_o)$ in the world-frame (Figure 2-4b). The hand is a line segment of length $2l$. The object, which is treated as a convex polygon, is parameterized by its (object-frame) vertex positions $\vec{v}_i = (x_i, y_i)$, outward facing surface normals $\hat{n}_i = (\cos\phi_i, \sin\phi_i)$, and contact face offsets $d_i = \hat{n}_i \cdot \vec{v}_i$ (Figure 2-4a). We denote the (world-frame) contact normals and contact tangents of the hand and ground as $(\hat{n}_h, \hat{t}_h)$ and $(\hat{n}_g, \hat{t}_g)$ respectively. The constant h_{grnd} represents the height of the ground. During flush contact with the hand, we also consider the relative tangential displacement between the hand and the object, s, where:

$$s = \hat{t}_h \cdot (\vec{r}_h - \vec{r}_o) \tag{2.2}$$

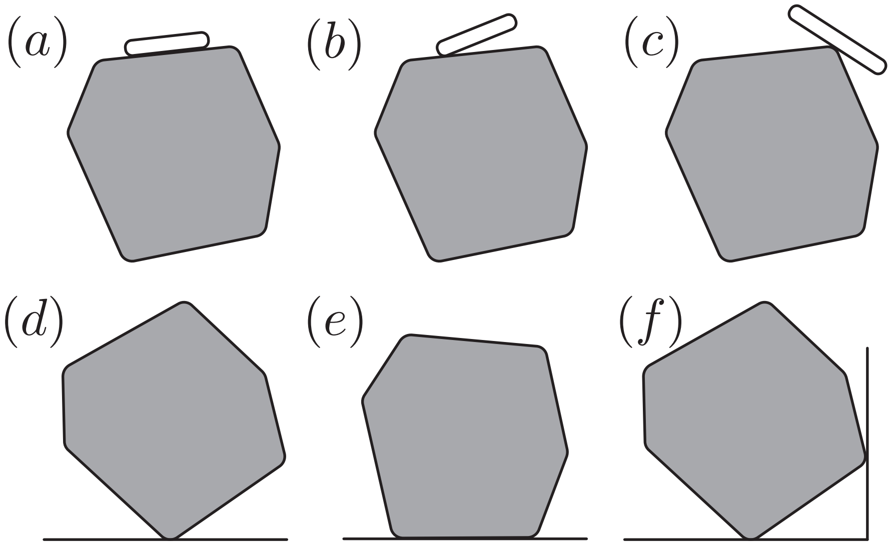


Figure 2-3: Different types of contact geometries between the object and hand/environment. (a) Flush contact between the hand and object. (b) Object-line/hand-point contact. (c) Object-point/hand-line contact. (d) Object-point/ground-line contact. (e) Flush contact between the object and ground. (f) Wall contact.

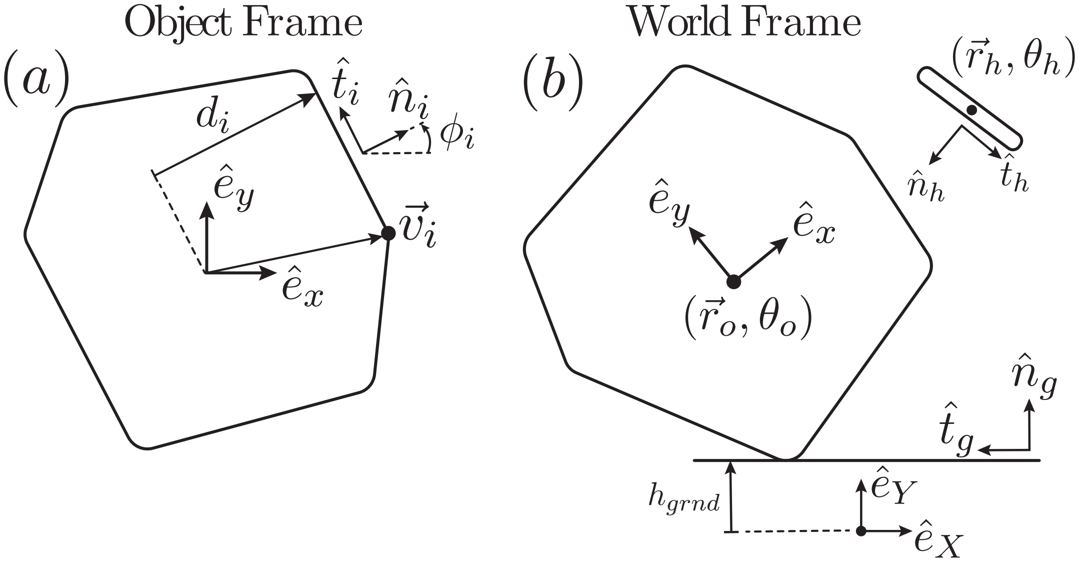


Figure 2-4: System parameterization.

When the object is in single point contact with the ground, we model the effect of gravity using the parameters (α_i, β_i), which encapsulate the object's weight (mg), the length (l_i) of the gravitational moment arm (the vector from the ground contact point to the object's

COM), and the angle $(\theta_o + \psi_i)$ of the gravitational moment arm:

$$\tau_{g,i} = mgl_i \sin(\theta_o + \psi_i) = \alpha_i \cos(\theta_o) + \beta_i \sin(\theta_o) \tag{2.3}$$

Where $\tau_{g,i}$ is the torque due to gravity w/respect to the “ith" vertex, which is assumed to be in contact with the ground. From the quasi-static motion assumption, it follows that:

$$\sum \vec{w}_{net} = \vec{w}_h + \vec{w}_e + \vec{w}_{grav} = 0, \tag{2.4}$$

where $\vec{w}_h = (\vec{F}_h, \tau_h)$, $\vec{w}_e = (\vec{F}_e, \tau_e)$, and $\vec{w}_{grav} = (-mg\hat{n}_g, 0)$ are the wrenches exerted on the object by the hand, environment, and gravity, respectively. Because the wrench exerted by gravity is constant, it follows that $\vec{w}_e = -\vec{w}_h + const$. This relation allows us to use the wrenches measured at the hand contact to infer the external contact behavior.

Measured vs. estimated quantities: We are able to directly measure the hand pose $(\vec{r}_h, \theta_h)$ and wrench $\vec{w}_e$ via robot proprioception and a force-torque sensor in the wrist of the robot. We have added limited visual feedback, which estimates the world-frame positions of the object vertices $\vec{v}_{i,world}$. Compared to the tactile feedback, vision is relatively noisy and operates at a lower frequency. We primarily use it to seed the estimator with an initial prior, though the estimator is also capable of making use of live visual feedback. Besides the hand pose and wrench, every other quantity, including the object vertices, is estimated.

2.2 An Introduction to Contact Mechanics

Before constructing the model of our system, it is important to describe the underlying physics that will govern the behavior of our system. Specifically, the questions we want to answer are:

- Which wrenches can be transmitted through contact?
- How does contact constrain the motion of a system?

To answer these questions, we will need to look at the geometry and frictional properties of the contact interface.

2.2.1 Contact Mode and the Friction Cone

In this work, when we use the term **contact mode**, we are specifically referring to whether or not a contact is sticking or slipping. The contact mode is determined by the contact forces and the friction properties of the contact.

We use the Coulomb friction law to model the effect of friction in our system. To understand this law, let's consider the example of a block that is in contact with the ground, as depicted in Figure 2-5 (left). Here, $\vec{F}_c$ is the reaction force of the ground acting on the block. We can decompose $\vec{F}_c$ into its normal and tangential components (w/respect to the ground), f_n and f_t. By the Coulomb friction law, f_n and f_t satisfy the following inequality:

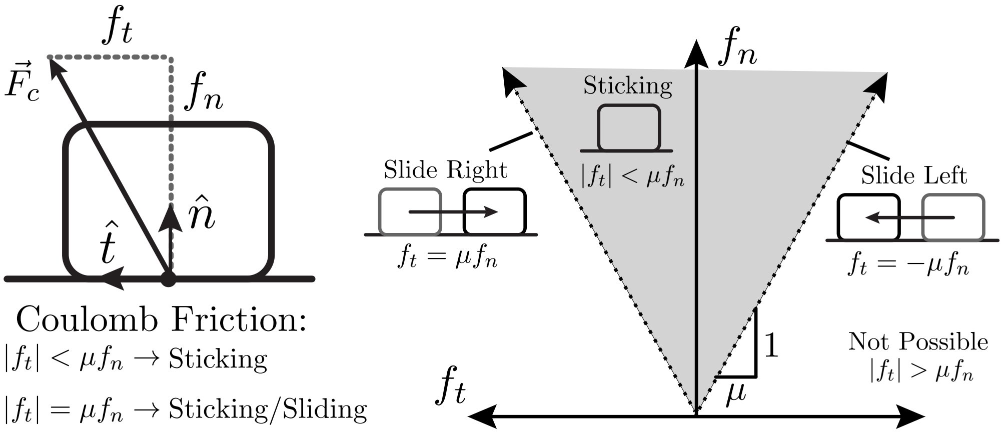


Figure 2-5: An example of the Coulomb friction model. Left: we consider a block that is in contact with the ground. The reaction force of the ground, $\vec{F}_c$, can be split into its normal and tangential components, f_n and f_t. Right: the set of possible reaction forces (which satisfy $|f_t| \leq \mu f_n$) is called the friction cone (shaded region). Reaction forces on the interior of the friction cone correspond to sticking contact, while forces on the boundary admit the possibility of sliding contact.

$$|f_t| \leq \mu f_n \tag{2.5}$$

We can rewrite this as a pair of linear constraints:

$$-\mu f_n + f_t \leq 0, \quad -\mu f_n - f_t \leq 0 \tag{2.6}$$

Note that $f_n \geq 0$ in this model, meaning that a surface can push, but cannot pull. The dimensionless parameter μ is called the static friction coefficient, and is determined by the pair of materials that are in contact with one another. The set of possible reaction forces (i.e. the set of reaction forces that satisfy (2.5)) is called the friction cone, which we depict as the shaded region in Figure 2-5 (right).

When $|f_t|$ is less than μf_n, the block is in sticking contact with the ground. However, when these two quantities are equal, the system admits the possibility of sliding (though sticking is also possible). In this case, we can check the sign of the tangential force f_t to determine which direction of sliding is admissible. Remember that $\vec{F}_c$ is the reaction force that the ground exerts on the block in order to resist the motion of the block. Thus, f_t points in the direction opposite of sliding: $f_t = \mu f_n$ means that the block is either sticking or sliding left, while $f_t = -\mu f_n$ means that the block is either sticking or sliding right.

Though we were using the interaction between the ground and a block as an example, we can use the Coulomb friction law to model any other contact interaction in our system. It should be noted that we only make use of the static friction model: we do not consider the change in the friction coefficient when a contact transitions between sticking and sliding. Though we do directly estimate the friction coefficient for the hand/object contact, this is not the case for the object/environment contact. Instead, our estimator builds an approximation of the friction cone itself by trying to identify its boundaries. This friction cone estimate is then used by the controller to determine which wrenches the robot needs to exert on the object in order to enforce sticking contact.

2.2.2 Contact Geometry and the Torque Cone

In the same way that the contact mode is determined by friction properties and the contact forces, the contact geometry (flush contact vs. point/line contact) is related to the contact wrench and the shapes of the two interacting bodies.

As before, let's consider the example of a block that is in contact with the ground, as depicted in Figure 2-6 (left). When the block and ground are flush with each other, the force that the ground exerts on the block is distributed across the contact patch. There are a few ways to model this. We could represent it as a force distribution (Figure 2-6 top left). Alternatively, we could choose a reference point (for instance the centroid of the contact patch, C), and compute both the net reaction force as well as the net resultant torque w/respect to this reference point (Figure 2-6 bottom left).

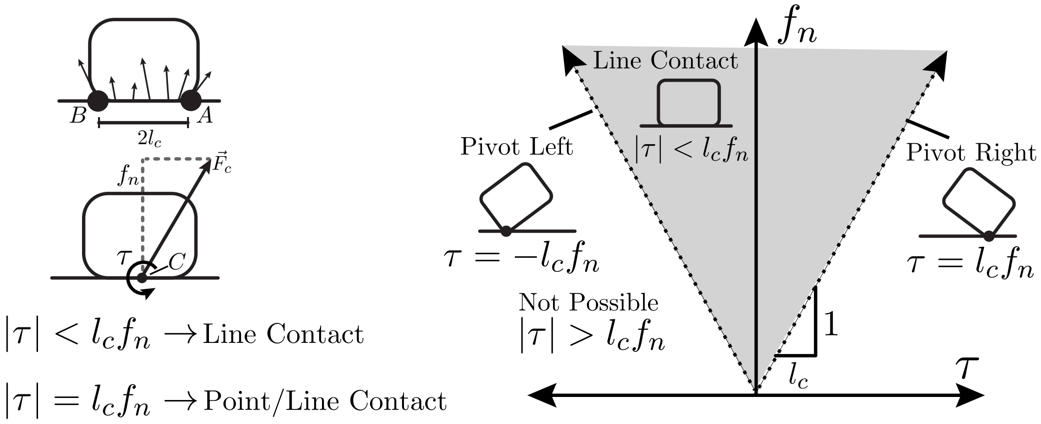


Figure 2-6: Left: we consider a block that is in contact with the ground. The reaction force of the ground $\vec{F}_c$ can be split into its normal and tangential components, f_n and f_t. We also consider the net resultant torque, τ, w/respect to the centroid of the contact patch (which has length $2l_c$). Right: the set of possible reaction wrenches (which satisfy $|\tau| \leq l_c f_n$) is called the torque cone (shaded region). Reaction wrenches on the interior of the torque cone correspond to flush contact, while wrenches on the boundary admit the possibility of point/line contact.

Let τ denote the resultant torque that the ground exerts on the block w/respect to C, and f_n denote the normal component of the net reaction force that the ground exerts on the block (as shown in Figure 2-6). The length of the contact patch is given by $2l_c$. To derive the torque constraints, let's look at the force distribution of the contact patch (Figure 2-7).

Here, the distributed contact force, $\vec{f}_c(x)$ has units of force over length (e.g., N/m) and is a function of the position of the point that its acting on, x. Remember that $\vec{f}_c(x)$ is referring to the reaction force of the ground acting on the block. The unit vectors $\hat{n}$ and $\hat{t}$ denote the surface normal and tangent respectively. Points A and B are the endpoints of the contact patch. Point D is an arbitrary point belonging to line segment AB, and l_D is the distance of D from A. Points E and F are just arbitrary points in space.

In this work, we always assume that a contact is not “sticking" i.e., that the contact

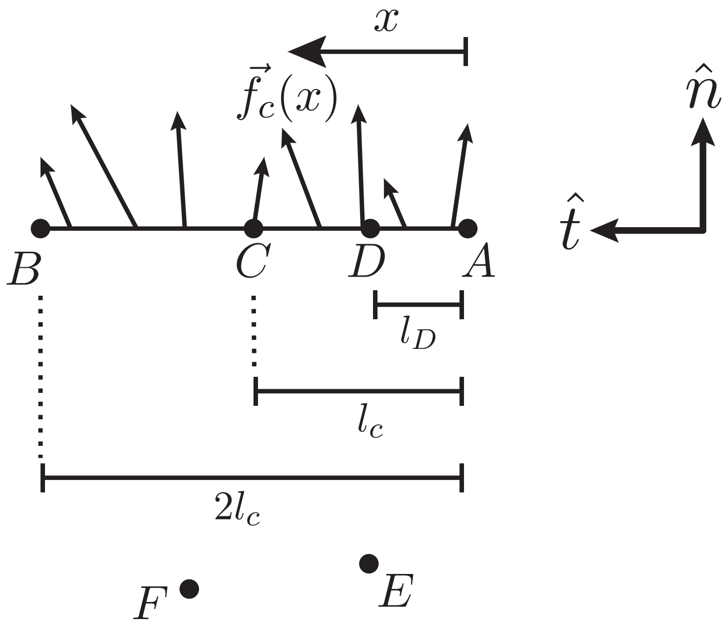


Figure 2-7: The distributed contact force.

patch can push but not pull. Mathematically speaking, what we mean is that:

$$\vec{f}_c(x) \cdot \hat{n} \geq 0 \tag{2.7}$$

In other words, the normal component of the distributed force is always non-negative. We can compute the total net contact force, $\vec{F}_c$, by integrating the distributed force across the contact patch:

$$\vec{F}_c = \int_0^{2l_c} \vec{f}_c(x)dx \tag{2.8}$$

We can now show that the normal component of $\vec{F}_c$ must also be non-negative ($f_n \geq 0$):

$$f_n = \vec{F}_c \cdot \hat{n} = \left(\int_0^{2l_c} \vec{f}_c(x)dx\right) \cdot \hat{n} = \int_0^{2l_c} \left(\vec{f}_c(x) \cdot \hat{n}\right) dx = \int_0^{2l_c} (q(x) \geq 0)\, dx \ \ \geq 0 \tag{2.9}$$

The resultant torque about D (which is any arbitrary point belonging to the contact patch) is given by:

$$\tau_D = \int_0^{2l_c} (\vec{r}(x) - \vec{r}_D) \times \vec{f}_c(x)dx = \int_0^{2l_c} (\vec{r}_A + x\hat{t} - \vec{r}_D) \times \vec{f}_c(x)dx \tag{2.10}$$

We are going to show that $-(2l_c - l_D)f_n \leq \tau_D \leq l_D f_n$. These are the linear torque constraints of the system. We begin by plugging in $\vec{r}_D = \vec{r}_A + l_D\hat{t}$ into equation (2.10):

$$\tau_D = \int_0^{2l_c} (\vec{r}_A + x\hat{t} - \vec{r}_A - l_D\hat{t}) \times \vec{f}_c(x)dx = \int_0^{2l_c} (x - l_D)\hat{t} \times \vec{f}_c(x)dx \tag{2.11}$$

We can now split up $\vec{f}_c$ into its normal and tangential components;

$$\tau_D = \int_0^{2l_c} (x - l_D)\hat{t} \times \Big((\vec{f}_c(x) \cdot \hat{n})\hat{n} + (\vec{f}_c(x) \cdot \hat{t})\hat{t}\Big)\, dx \tag{2.12}$$

Since $\hat{t} \times \hat{t} = 0$ and $\hat{t} \times \hat{n} = -1$, the above simplifies to:

$$\tau_D = \int_0^{2l_c} (l_D - x)(\vec{f}_c(x) \cdot \hat{n})dx \tag{2.13}$$

$$\tau_D = \int_0^{l_D} (l_D - x)(\vec{f}_c(x) \cdot \hat{n})dx + \int_{l_D}^{2l_c} (l_D - x)(\vec{f}_c(x) \cdot \hat{n})dx \tag{2.14}$$

$$\tau_D = \int_0^{l_D} (l_D - x)(\vec{f}_c(x) \cdot \hat{n})dx - \int_{l_D}^{2l_c} (x - l_D)(\vec{f}_c(x) \cdot \hat{n})dx \tag{2.15}$$

On the interval $x \in [0, l_D]$, we see that $0 \leq l_D - x \leq l_D$. Since $\vec{f}_c(x) \cdot \hat{n} \geq 0$, it follows that:

$$\int_0^{l_D} (l_D - x)(\vec{f}_c(x) \cdot \hat{n})dx = \int_0^{l_D} (p(x) \geq 0)(q(x) \geq 0)dx \quad \geq 0 \tag{2.16}$$

$$\int_0^{l_D} (l_D - x)(\vec{f}_c(x) \cdot \hat{n})dx \leq l_D \int_0^{l_D} (\vec{f}_c(x) \cdot \hat{n})dx \leq l_D \int_0^{2l_c} (\vec{f}_c(x) \cdot \hat{n})dx = l_D f_n \tag{2.17}$$

$$0 \leq \int_0^{l_D} (l_D - x)(\vec{f}_c(x) \cdot \hat{n})dx \leq l_D f_n \tag{2.18}$$

We can use a similar set of steps to show that:

$$0 \leq \int_{l_D}^{2l_c} (x - l_D)(\vec{f}_c(x) \cdot \hat{n})dx \leq (2l_c - l_D)f_n \tag{2.19}$$

Plugging these inequalities into (2.15), we get our result:

$$-(2l_c - l_D)f_n \leq \tau_D \leq l_D f_n \tag{2.20}$$

In the case of the endpoints A and B, it follows that:

$$-2l_c \leq \tau_A \leq 0 \leq \tau_B \leq 2l_c f_n \tag{2.21}$$

For the specific case of the centroid of the contact patch, C, we see that:

$$-l_c f_n \leq \tau \leq l_c f_n \quad \rightarrow \quad |\tau| \leq l_c f_n \tag{2.22}$$

$$-l_c f_n + \tau \leq 0, \qquad -l_c f_n - \tau \leq 0 \tag{2.23}$$

The set of possible pairs of f_n and τ that satisfy (2.22) is called the torque cone, which we depict as the shaded region in Figure 2-6 (right). Note how similar (2.22) and (2.23) are to (2.5) and (2.6). The intersection of the friction and torque cones define a structure called **the generalized friction cone** (or the wrench cone) [3, 17].

When $|\tau|$ is less than $l_c f_n$, the block is in flush contact with the ground. However, when these two quantities are equal, the system admits the possibility of pivoting about one of the two endpoints of the contact patch (though flush contact is also possible). To determine which of these two endpoints is the pivot location, we can check the sign of the resultant torque. The equality $\tau = -l_c f_n$ corresponds to all of the contact force being concentrated at B (generating a clockwise moment about C), and thus admits the possibility of pivoting about the left endpoint. The equality $\tau = l_c f_n$, on the other hand, corresponds to all of the contact force being concentrated at A (generating a counterclockwise moment about C), and thus admits the possibility of pivoting about the right endpoint. These two possibilities (flush contact and pivoting) correspond to the two types of contact geometries (flush contact and point/line contact).

One last property that is useful to show is the relationship between the resultant torques about two different points. Consider points E and F as shown in Figure 2-7. Their corresponding resultant torques are given by:

$$\tau_E = \int_0^{2l_c} (\vec{r}(x) - \vec{r}_E) \times \vec{f}_c(x)dx \tag{2.24}$$

$$\tau_F = \int_0^{2l_c} (\vec{r}(x) - \vec{r}_F) \times \vec{f}_c(x)dx \tag{2.25}$$

If we take their difference, we see that:

$$\tau_F - \tau_E = \int_0^{2l_c} (\vec{r}(x) - \vec{r}_F) \times \vec{f}_c(x)dx - \int_0^{2l_c} (\vec{r}(x) - \vec{r}_E) \times \vec{f}_c(x)dx \tag{2.26}$$

$$\tau_F - \tau_E = \int_0^{2l_c} (\vec{r}_E - \vec{r}_F) \times \vec{f}_c(x)dx = (\vec{r}_E - \vec{r}_F) \times \int_0^{2l_c} \vec{f}_c(x)dx \tag{2.27}$$

$$\tau_F - \tau_E = (\vec{r}_E - \vec{r}_F) \times \vec{F}_c \tag{2.28}$$

$$\tau_F = \tau_E + (\vec{r}_E - \vec{r}_F) \times \vec{F}_c \tag{2.29}$$

Equation (2.29) allows us to update the resultant torque when changing its reference point.

2.2.3 Kinematic and Wrench Constraints of Two Contacting Bodies

The example of a block interacting with the ground is great for building intuition for how contact constraints work. Let's now apply this understanding to a more general set of interactions. Consider two contacting bodies, depicted in Figure 2-8. Here, A and D are material points fixed to the first body frame, while B and E are material points fixed to the second body frame. The unit vectors $(\hat{n}, \hat{t})$ are the normal and tangent of the first body contact face. The wrench $\vec{w} = (f_n\hat{n} + f_t\hat{t}, \tau)$ is the net contact wrench that the second body exerts on the first body, measured w/respect to reference point C, which is fixed in the first body frame. The friction coefficient for this contact is given by μ. We consider point/line contact (Figure 2-8a), flush contact (Figure 2-8b), and no contact (not depicted). We will now assemble a set of kinematic and wrench constraints for each of these configurations.

Making and breaking contact: The contact wrench can only be nonzero during contact. During periods of no contact, the contact wrench is zero. For the interaction in Figure 2-8:

$$f_n > 0 \text{ implies } (\vec{r}_B - \vec{r}_A) \cdot \hat{n} = 0 \text{ and/or } (\vec{r}_E - \vec{r}_A) \cdot \hat{n} = 0 \tag{2.30}$$

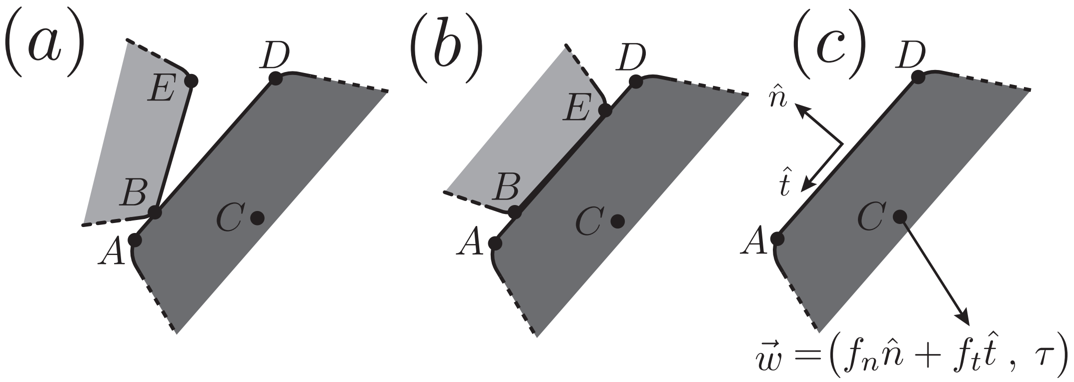


Figure 2-8: a) Point/line contact. b) Flush contact. c) Normal/tangent vectors.

where $(\vec{r}_B - \vec{r}_A) \cdot \hat{n}$ and $(\vec{r}_E - \vec{r}_A) \cdot \hat{n}$ are the respective signed distances of points B and E from the line defined by AD.

Sticking, sliding, and the friction cone: Let's generate the kinematic and wrench constraints associated with friction for both configurations depicted in Figure 2-8. We will assume without loss of generality that B is in contact with line segment AD for both of these configurations (as depicted), though point/line contact could be achieved by having point E contact segment AD instead. The tangential velocity of B w/respect to segment AD is given by $v_{slide} = \frac{d}{dt}((\vec{r}_b - \vec{r}_a) \cdot \hat{t})$, where $v_{slide} > 0$ corresponds to sliding to the left, $v_{slide} < 0$ corresponds to the right, and $v_{slide} = 0$ corresponds to sticking. Combining this with the Coulomb friction law, we get:

$$v_{slide} = \frac{d}{dt}((\vec{r}_b - \vec{r}_a) \cdot \hat{t}) = v^+_{slide} - v^-_{slide} \tag{2.31}$$

$$0 \leq -\mu f_n - f_t \quad \perp \quad v^+_{slide} \geq 0 \tag{2.32}$$

$$0 \leq -\mu f_n + f_t \quad \perp \quad v^-_{slide} \geq 0 \tag{2.33}$$

The $\perp$ symbol is a shorthand, where $0 \leq u \perp w \geq 0$ implies $u \geq 0$, $w \geq 0$, and $uw = 0$. Equations (2.32) and (2.33) are called complementarity constraints. Essentially, (2.32) and (2.33) state that point B can only slide left (relative to the first body) if $f_t = -\mu f_n$, that B can only slide right if $f_t = \mu f_n$, and that B is sticking when the reaction force is on the interior of the friction cone. When checking the signs in front of each of the terms remember that

$\vec{w} = (f_n\hat{n} + f_t\hat{t}, \tau)$ is the contact wrench that the second body BE exerts on the first body AD. This is the opposite from the block/ground example, in which we consider the wrench that the ground exerts on the block.

Contact geometry and torque constraints: When generating the kinematic and wrench constraints associated with the contact geometry, it is convenient to first define the resultant contact torque, τ_P, about some reference point P. Plugging points P and C into equation (2.29), we get:

$$\tau_P = \tau + (\vec{r}_C - \vec{r}_P) \times (f_n\hat{n} + f_t\hat{t}) \tag{2.34}$$

As we move forward, one thing to keep in mind is that a force generates zero resultant torque w/respect to the point that the force is acting at. This is because the corresponding moment arm has a length of zero.

When the system is in point/line contact at B (Figure 2-8a), the contact force is transmitted through point B. Thus, the resultant contact torque about B must be zero. Moreover, because B is on the interior of segment AD in this configuration, it follows that the resultant torque about A is clockwise, and the resultant torque about D is counterclockwise:

$$\tau_A \leq 0, \;\; \tau_B = 0, \;\; \tau_E \geq 0 \;\; \text{(point/line contact at } B\text{)} \tag{2.35}$$

Equation (2.35) is simply a different way to express the torque cone constraints; one that depends on the positions of the various reference points instead of the lengths of the contacting faces.

In the case of flush contact, the contact force is distributed across the contact patch instead of being concentrated at a single point. Applying equation (2.21) to the endpoints of the two contacting faces, we see that:

$$\tau_A \leq 0, \;\; \tau_B \leq 0, \;\; \tau_D \geq 0, \;\; \tau_E \geq 0 \;\; \text{(flush contact)} \tag{2.36}$$

Note that this statement holds true as long as there is any amount of overlap between segments AD and BE. Segment AD could entirely contain BE (or vice versa) and this would still be true.

These torque constraints allow us to disambiguate contact geometries, estimate contact locations, and regulate the contact geometry. For these purposes, it is convenient to consider the center of pressure (COP) of a line contact defined by points G and H e.g., the point on the contact patch for which the resultant contact torque is zero:

$$\vec{r}_Q = COP(\vec{r}_G, \vec{r}_H) \iff \tau_Q = 0, \vec{r}_Q = \gamma\vec{r}_G + (1-\gamma)\vec{r}_H \tag{2.37}$$

In other words, point Q is the COP of the line passing through G and H if both Q lives on that line, and the resultant torque about Q is zero. For the point contact illustrated in Figure 2-8a, B is the COP of AD. During flush contact, the COP belongs to the intersection of segments AD and BE. This behavior is illustrated in Figure 2-9

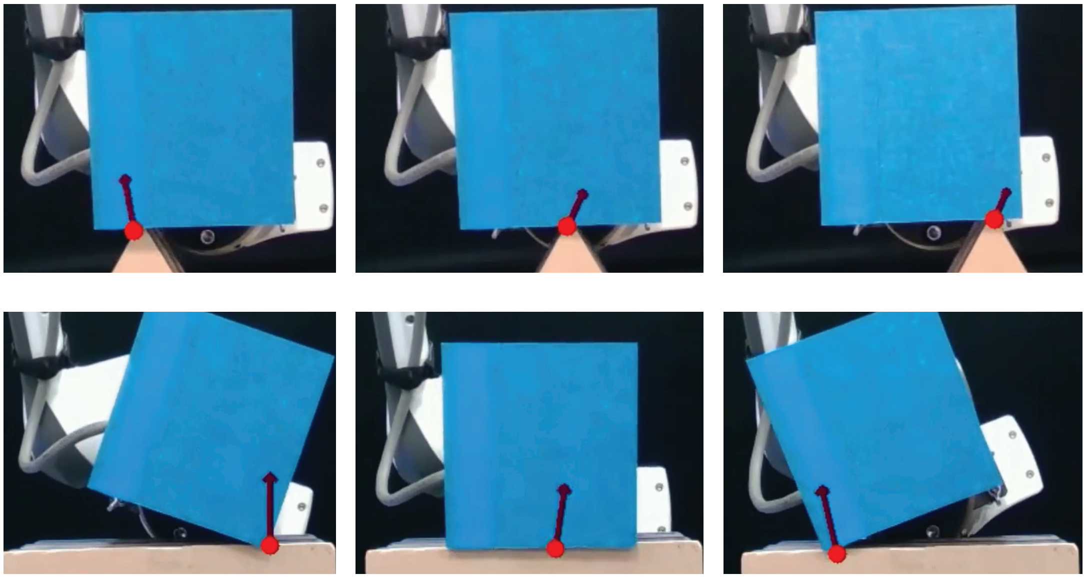


Figure 2-9: Experiments demonstrating the relationship between the center of pressure (COP) of a contact patch and the contact geometry. The location of the COP (red dot) is computed using the measured contact wrench and the known pose/geometry of the blue square. **Top row:** During point/line contact with the brown triangle, the COP always coincides with the contact location. **Bottom row:** In this experiment, the square is in contact with a large surface (brown), whose boundaries extend far in both directions. During point/line contact with this surface (right/left), the COP coincides with the contact location. During flush contact (center), the COP is on the interior of the contact patch.

During estimation, we frequently solve (2.37) to compute the COP of a potential contact patch, given its endpoints and the measured contact wrench. This information can then be used to help predict the current contact geometry.

2.2.4 Admissible Motions, Constraints, and the Kinematics of Contact

The mode (sticking/slipping) and geometry (point/line vs. flush contact) at each of the contacts impose a set of kinematic constraints, restricting the set of motions available to the system. The admissible motion directions are the remaining degrees of freedom, and depend on the current contact configuration of the system. We describe the admissible motion directions using the corresponding tangent space vectors for each of the available degrees of freedom.

However, before we take a look at the math, let's describe the kinematic constraints of contact qualitatively. We'll start by looking at the most restrictive type of contact (flush sticking contact), and then see how removing the flush or sticking constraints add degrees of freedom to the system.

During flush contact, the respective contact faces of each of the bodies must be collinear. This means that the contact faces must have the same orientation, and cannot move relative to each other in the normal direction. The sticking constraint restricts the relative motion of the contact faces in the tangential direction. Thus, in flush sticking contact, the two contacting bodies are constrained to move together as a single rigid body (Figure 2-10). Depending on the other constraints on the bodies, the system may still have up to three degrees of freedom or may not have any remaining at all.

If we allow for sliding motion, we gain an extra degree of freedom corresponding to relative motion of the contact faces in the tangential direction (Figure 2-11 top row). If instead we allow for point/line contact, we gain an extra degree of freedom corresponding to relative rotation between the two contacting bodies (Figure 2-11 bottom row).

To understand how we arrive at the various admissible motion directions, let's take a look once again at the example of a block that is in contact with the ground (Figure 2-12). We'll consider the case of no contact, as well as the different combinations of sticking/sliding and point/line vs. flush contact. In this example, we will be expressing the

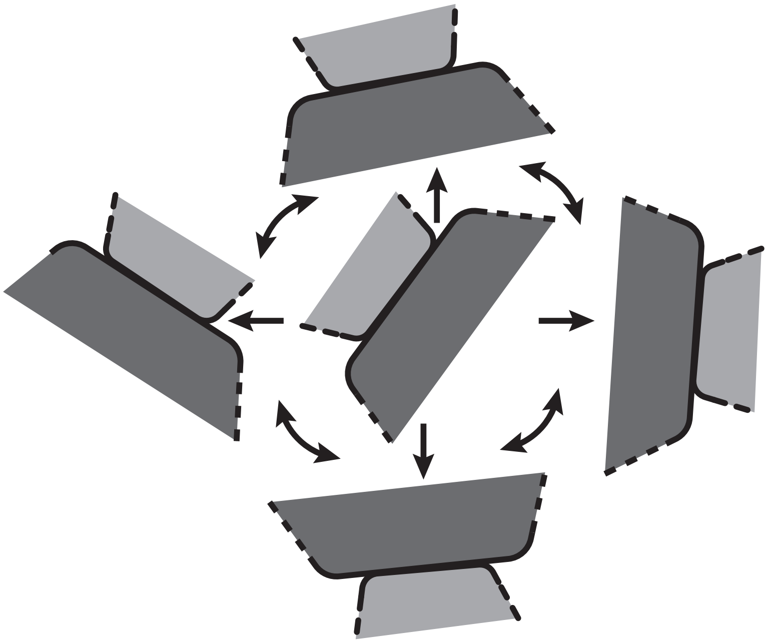


Figure 2-10: During flush sticking contact, the two contacting bodies are constrained to move together as a single rigid body.

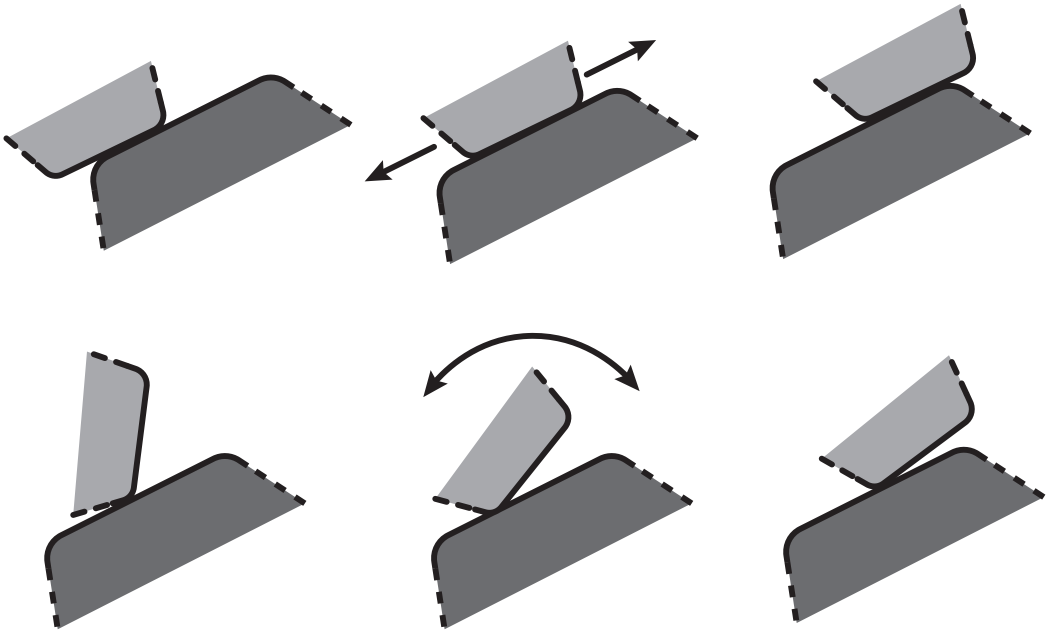


Figure 2-11: **Top row:** Sliding allows for relative motion between the two bodies in the tangential direction. **Bottom row:** Point contact allows for relative rotation between the two bodies.

admissible motion directions in terms of the object's pose. However, in our work we usually describe the admissible motion direction of the system in terms of the hand pose.

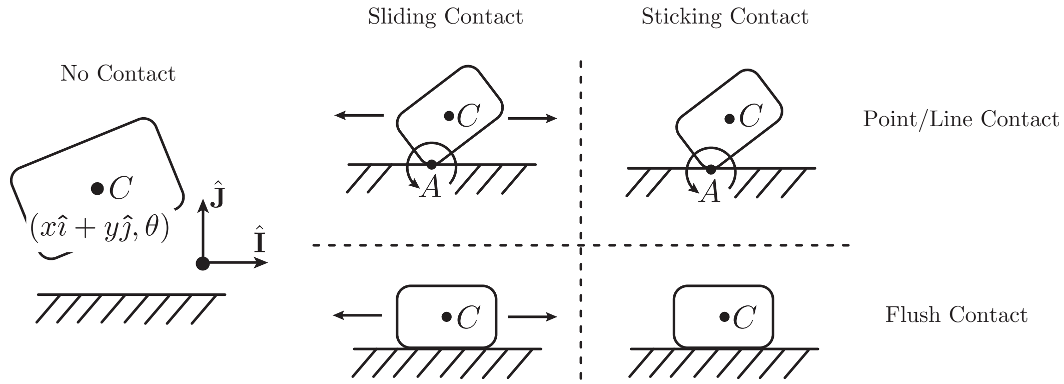


Figure 2-12: A block in various discrete contact states with the ground. The pose of the block is given by $(\vec{r}_C, \theta)$, where $\vec{r}_C = x\hat{\mathbf{I}} + y\hat{\mathbf{J}}$ is the position of its centroid, and θ is its orientation. We look at the case of no contact, as well as the different combinations of sticking/sliding and point/line vs. flush contact.

No contact: In the case of no contact, the block is free to translate and rotate in the plane without constraint. In this case, the admissible motion directions are given by:

$$\Delta\vec{q}_x = (\hat{\mathbf{I}}, 0), \;\; \Delta\vec{q}_y = (\hat{\mathbf{J}}, 0), \;\; \Delta\vec{q}_\theta = (0, 1) \tag{2.38}$$

where $\Delta\vec{q}_x$ and $\Delta\vec{q}_y$ correspond to pure horizontal and vertical translation, and $\Delta\vec{q}_\theta$ corresponds to pure rotation about the centroid. It should be noted that we could construct a different set of admissible motion directions for this configuration; they just need to be linearly independent and have the same span.

Sliding point/line contact: In the case of sliding point/line contact, the block has two degrees of freedom. It can pivot about the contact point A, or it can slide along the ground. Thus, the corresponding admissible motion directions are given by:

$$\Delta\vec{q}_x = (\hat{\mathbf{I}}, 0), \;\; \Delta\vec{q}_\theta = (\hat{\mathbf{K}} \times (\vec{r}_C - \vec{r}_A), 1) \tag{2.39}$$

where $\Delta\vec{q}_x$ corresponds to pure horizontal translation, and $\Delta\vec{q}_\theta$ corresponds to pure rotation about the contact point A.

Sticking point/line contact: In the case of sticking point/line contact, the block can only pivot about the contact point A. Thus, the only admissible motion direction is given by:

$$\Delta\vec{q}_\theta = (\hat{\mathbf{K}} \times (\vec{r}_C - \vec{r}_A), 1) \tag{2.40}$$

Sliding flush contact: In the case of sliding flush contact, the block can only sliding along the ground. Thus, the only admissible motion direction is given by:

$$\Delta\vec{q}_x = (\hat{\mathbf{I}}, 0) \tag{2.41}$$

Sticking flush contact: In the case of sticking flush contact, the motion of the block is fully constrained, and there are no admissible motions.

2.3 System Model

Now that we've discussed the general principles that govern the behavior of our system, we can now build a model to describe this particular system's kinematics and dynamics. This analysis will be split into three parts: friction cones, torque cones, and kinematic constraints/admissible motion directions. We will be using the parameterization described in Section 2.1.3

2.3.1 Hand and Ground Friction Cones

Here, we will construct the set of wrench-space constraints that describe sticking at both the hand and ground contacts. In this derivation, we'll assume that the object is in contact with the hand and the ground, but not any walls. Let's begin by taking a look at the free body diagram of the system (Figure 2-13). There are wrenches acting on the object: the hand contact wrench, $\vec{w}_h = (\vec{F}_h, \tau)$, which is taken w/respect to the robot TCP, the ground wrench ,$\vec{w}_g = (\vec{F}_g, 0)$, which is taken w/respect to ground contact point (during point/line contact) or the centroid of the contact patch (during flush contact), and gravity $(-mg\hat{\mathbf{J}}, 0)$.

The force exerted by the hand can be split into its normal and tangential components, F_{hn} and F_{ht}. Similarly, we will split the force exerted by the ground into its horizontal and vertical components F_{gx} and F_{gy}. The friction coefficient of the hand and ground contacts are given by μ_h and μ_g respectively.

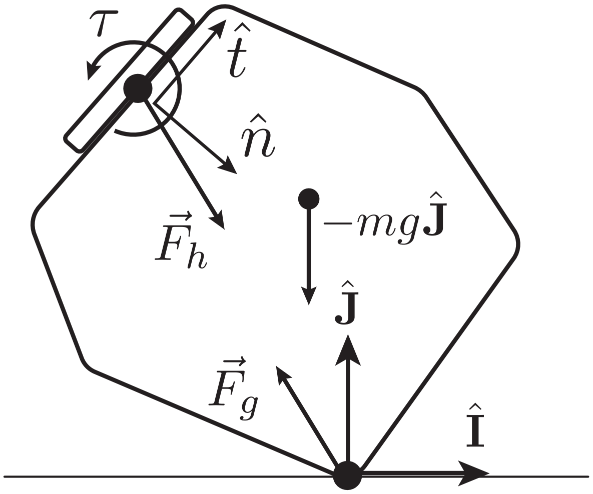


Figure 2-13: Free body diagram of the object.

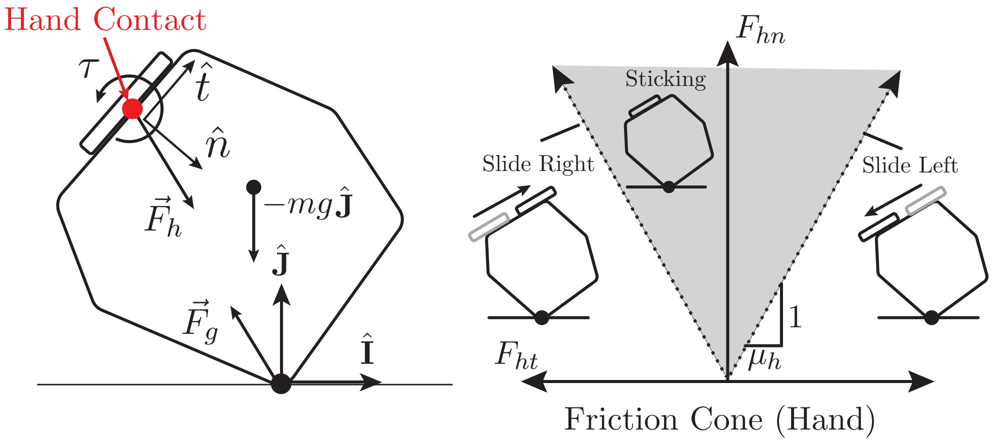


Figure 2-14: Friction cone describing the sticking/sliding conditions of the hand contact.

We can now apply what we've learned to construct the set of friction constraints for the system. We'll begin with the friction cone of the hand contact, as shown in Figure 2-14.

This corresponds to the wrench constraints:

$$-\mu F_{hn} - F_{ht} \leq 0 \;\; \rightarrow \;\; \vec{w}_h \cdot \vec{n}_{h^+} \leq 0 \tag{2.42}$$

$$-\mu F_{hn} + F_{ht} \leq 0 \;\; \rightarrow \;\; \vec{w}_h \cdot \vec{n}_{h^-} \leq 0 \tag{2.43}$$

where the wrench vectors $\vec{n}_{h^+}$ and $\vec{n}_{h^-}$ are the boundary normals of the hand friction cone. Note that the friction cone for the hand contact is fixed to the end-effector frame. This property will be useful later as we develop our friction constraint estimator.

The friction cone of the ground contact, as shown in Figure 2-15 (top right), corresponds to the wrench constraints:

$$-\mu F_{gy} - F_{gx} \leq 0, \quad -\mu F_{gy} + F_{ht} \leq 0 \tag{2.44}$$

Note that this expressed in terms of the force exerted by the ground $\vec{F}_g$. To express it in terms of the force exerted by the hand, we apply the quasi-static assumption:

$$\vec{F}_h + \vec{F}_g - mg\hat{\mathbf{J}} = 0 \;\; \rightarrow \;\; \vec{F}_h = mg\hat{\mathbf{J}} - \vec{F}_g \tag{2.45}$$

$$0 \leq -\mu_g (F_{hy} - mg) - F_{hx} \tag{2.46}$$

$$0 \leq -\mu_g (F_{hy} - mg) + F_{hx} \tag{2.47}$$

This transformation flips the friction cone upside down, and translates it upwards by mg (Figure 2-15 bottom). Note that the friction cone for the ground contact, even when expressed in terms of the force exerted by the hand, is fixed to the world frame. This property will also be useful for when we build our friction constraint estimator.

We now see that the friction cones for the hand and ground contacts can be expressed as linear constraints on the wrench exerted by the hand. Currently however, these constraints are expressed using two different basis: the world $(\hat{\mathbf{I}}, \hat{\mathbf{J}})$ and end-effector $(\hat{n}, \hat{t})$. If the orientation of the end-effector is known, we can rotate the ground friction cone so that it is expressed using the end-effector basis (as shown in Figure 2-16).

$$\vec{w}_h \cdot \vec{n}_{g^+} \leq b_{g^+} \tag{2.48}$$

$$\vec{w}_h \cdot \vec{n}_{g^-} \leq b_{g^-} \tag{2.49}$$

Here, $\vec{n}_{g^+}$ and $\vec{n}_{g^-}$ are the constraint boundary normals of the ground contact friction cone. The constants b_{g^+} and b_{g^-} are the constraint offsets.

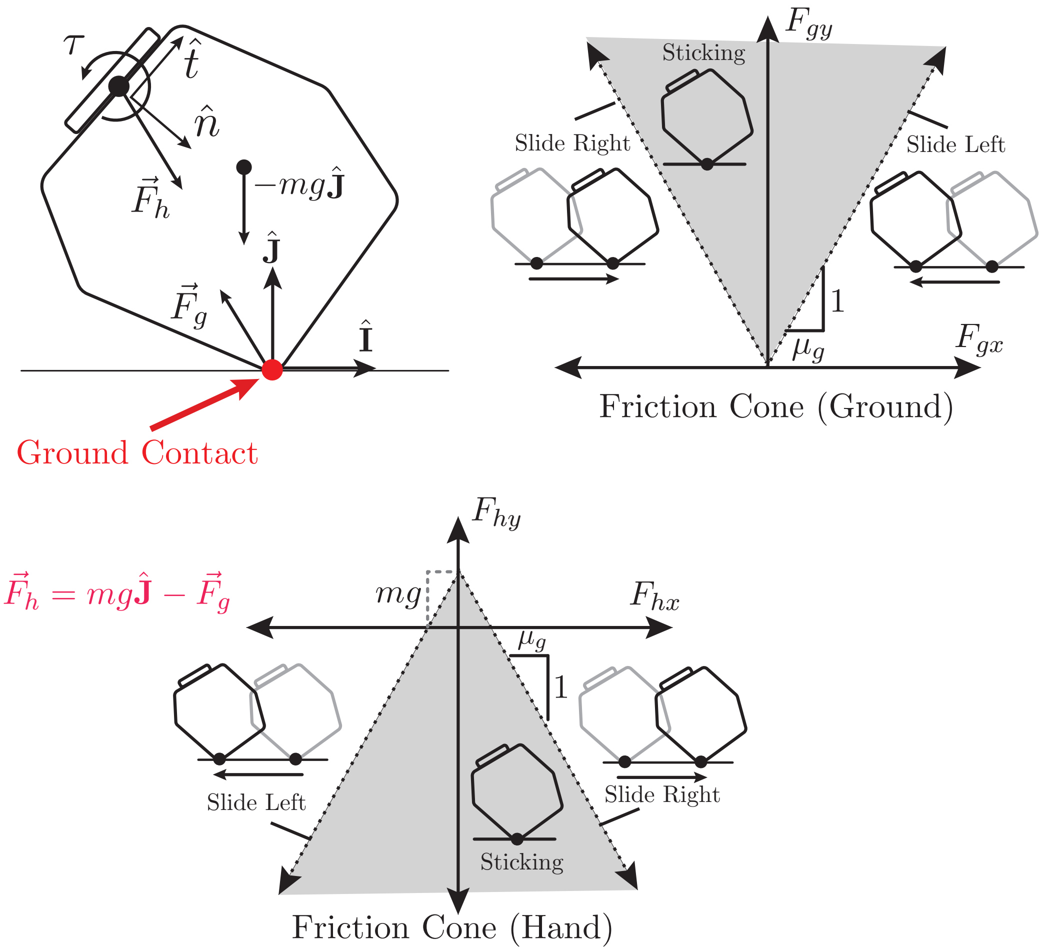


Figure 2-15: The friction cone of the ground contact expressed in terms of the force exerted by the ground (top right), and the force exerted by the hand (bottom)

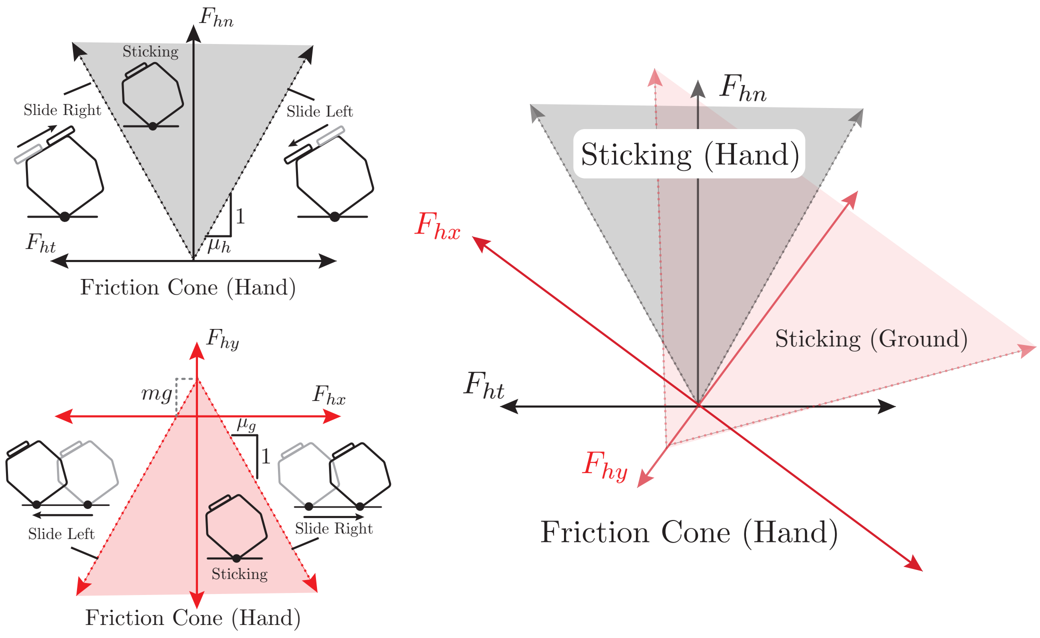


Figure 2-16: We can combine the friction cones of the hand and ground contacts to get a set of wrench constraints describing the conditions necessary to enforce sticking at both contacts.

2.3.2 Hand and Ground Torque Cones

We will now derive the set of wrench space constraints describing point/line and flush contact at the hand and environment contacts. Here, we'll assume that the object is in contact with the hand, and that either one vertex or two adjacent vertices of the object are in contact with the environment.

We'll start by looking at the constraints associated with the hand contact (Figure 2-17). The hand is modeled as a line segment of length $2l$. The vectors $\hat{n}$ and $\hat{t}$ are the unit normal and tangent of the hand respectively. The wrench that the hand exerts on the object $(\vec{F}_h, \tau)$ is measured w/respect to the centroid of the hand. The force exerted by the hand, $\vec{F}_h$ can be decomposed into its normal and tangential components, F_{hn} and F_{ht}. During flush contact, we consider the distances l_a, l_b of the left and right boundaries of the object face from the hand centroid. During point/line contact, we consider the contact point A, and its signed distance from the hand centroid, x_{COP}.

From the analysis in Section 2.2.2, we see that there are two pairs of inequalities associated with flush contact between the hand and object. The first pair of inequalities depends upon the length of the hand (Figure 2-17 top):

$$-lF_{hn} \leq \tau \leq lF_{hn} \tag{2.50}$$

The second pair of inequalities depends upon the locations of object vertices relative to the hand (Figure 2-17 middle):

$$-l_a F_{hn} \leq \tau \leq l_b F_{hn} \tag{2.51}$$

Both of these inequalities are true during flush contact. During the initial development of our controller, we assumed there was no "overhang" (i.e. the hand contact face never extended past the endpoints of the object face). Under this assumption, $l \leq l_a, l_b$, meaning that it was sufficient to use the length of the hand (which is known) for regulating the contact geometry at the hand. Once we developed our estimator to the point where it could keep track of the object vertex positions, it was then possible to estimate l_a and l_b, which allowed for a more accurate estimate of the torque cone constraints.

During object-point/hand-line contact, the center of pressure of the hand is coincident with the contact location, A, (Figure 2-17 bottom). In this case, the torque cone collapses into a single equality:

$$\tau = -x_{COP} F_{hn} \tag{2.52}$$

where x_{COP} is the signed distance of the center of pressure, A, from the hand centroid.

We turn our focus to the object/environment contact. In the case of flush contact (Figure 2-18), we assume a massless object. Combining this with the quasi-static assumption, it follows that the hand and ground contact wrenches must be equal and opposite: $\vec{w}_h = -\vec{w}_g$. This allows us to convert the torque constraints on the ground contact wrench into constraints on the hand contact wrench. If we apply equation (2.21) to the endpoints of the

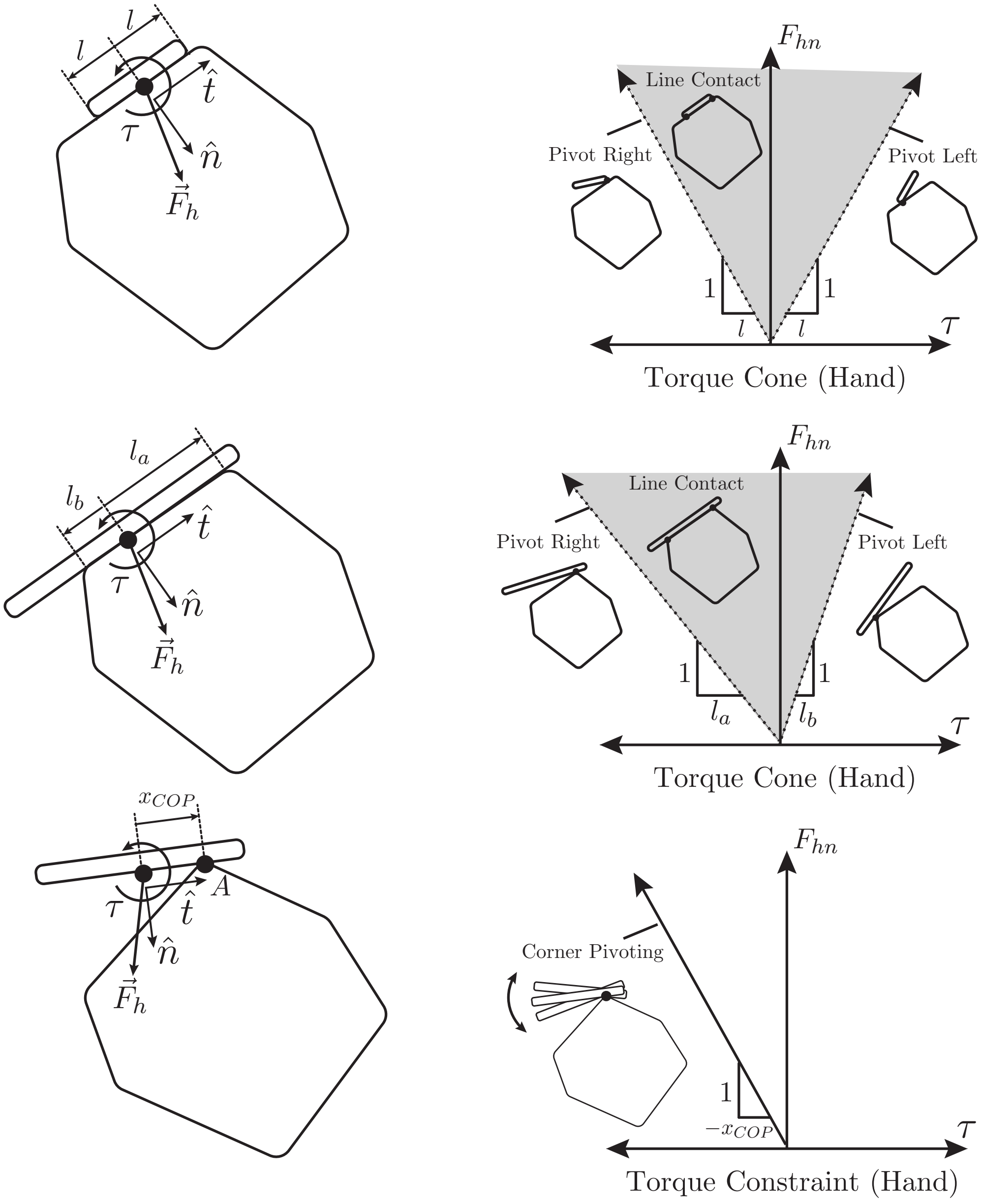


Figure 2-17: The torque constraints for the hand/object contact. **Top row:** When the hand contact face is contained within the object face, the torque cone constraints are determined by the length of the hand. **Middle row:** In the case of “overhang" the torque cone boundary is determine by the distance from the hand centroid to corresponding object vertex. **Bottom row:** During object corner pivoting, the object vertex *A* is also the center of pressure for the hand contact. Here, the torque cone reduces to a line, whose slope is a function of the distance of the COP from the hand centroid.

contact patch, A and B, we see that

$$\tau_B \leq 0 \leq \tau_A \tag{2.53}$$

where τ_A and τ_B are the resultant torques that the hand exerts about A and B respectively:

$$\tau_A = \tau + (\vec{r}_C - \vec{r}_A) \times \vec{F}_h \tag{2.54}$$

$$\tau_B = \tau + (\vec{r}_C - \vec{r}_B) \times \vec{F}_h \tag{2.55}$$

The equality conditions $\tau_A = 0$ and $\tau_B = 0$ admit the possibility of pivoting about points A and B respectively. The center of pressure of the ground contact patch, Q, lives somewhere on segment AB:

$$\vec{r}_Q = \frac{l-x}{l}\vec{r}_A + \frac{x}{l}\vec{r}_B \tag{2.56}$$

$$\tau_Q = \frac{l-x}{l}\tau_A + \frac{x}{l}\tau_B \tag{2.57}$$

Assuming a massless object, the resultant torque of the hand w/respect to Q is zero:

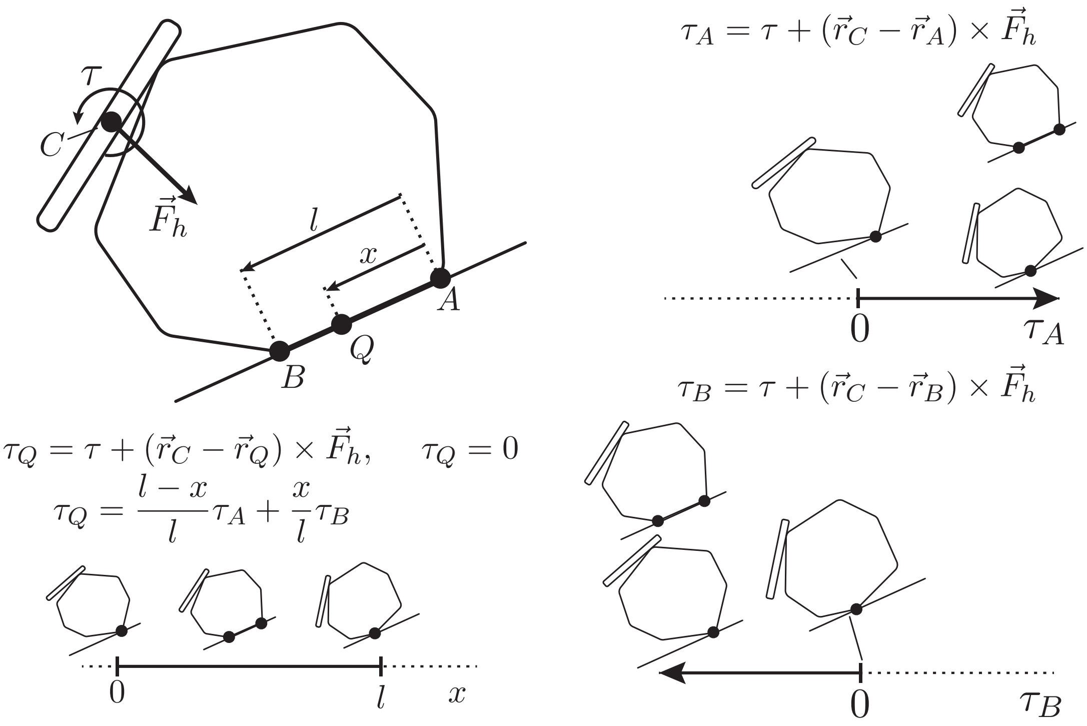
$$\tau_Q = \tau + (\vec{r}_Q - \vec{r}_A) \times \vec{F}_h, \qquad \tau_Q = 0 \tag{2.58}$$

When Q coincides with A, the system admits the possibility of pivoting about A. Similarly, when Q coincides with B, the system admits the possibility of pivoting about B. Finally, if Q is strictly on the interior of segment AB, the flush contact between the object and environment is enforced.

Finally, let's look at the case of point/line contact between the object and ground (Figure 2-19). Unlike flush contact, we will take the effect of gravity into account. The torque due to gravity taken w/respect to the pivot point, Q, is given by:

$$\tau_{gravity} = (\vec{r}_D - \vec{Q}) \times -mg\hat{J} \tag{2.59}$$

where m and D are the mass of the object and center of mass location respectively. Since,



τ
C
$\vec{F}_h$
l
x
A
Q
B
$\tau_A = \tau + (\vec{r}_C - \vec{r}_A) \times \vec{F}_h$
0
τ_A
$\tau_B = \tau + (\vec{r}_C - \vec{r}_B) \times \vec{F}_h$
$\tau_Q = \tau + (\vec{r}_C - \vec{r}_Q) \times \vec{F}_h, \quad \tau_Q = 0$
$\tau_Q = \frac{l-x}{l}\tau_A + \frac{x}{l}\tau_B$
0
l
x
0
τ_B

Figure 2-18: The torque constraints for the object/environment interface during flush contact. **Top left:** Partial free body diagram of the system. During flush contact, we assume that the object is massless, so only the hand and the environment exert wrenches on the object (environment wrench not pictured). **Top right:** The resultant torque, τ_A, that the hand exerts about the right endpoint, A, must be greater than or equal to zero. When $\tau_A = 0$, the system admits the possibility of pivoting about A. **Bottom right:** The resultant torque, τ_B, that the hand exerts about the left endpoint, B, must be less than or equal to zero. When $\tau_B = 0$, the system admits the possibility of pivoting about B. **Bottom left:** The resultant torque, τ_Q that the hand exerts about the center of pressure of the contact patch, Q, is zero (assuming the object is massless). When Q coincides with A, the system admits the possibility of pivoting about A. Similarly, when Q coincides with B, the system admits the possibility of pivoting about B.

during point/line contact, the contact location is a fixed vertex in the object frame, we can rewrite (2.59) as follows:

$$\tau_{gravity} = -mgl_i \sin(\theta_{obj} + \Delta\theta_i) = \alpha_i \cos\theta_{obj} + \beta_i \sin\theta_{obj} \tag{2.60}$$

$$\tau_{gravity} = \alpha_i \cos\theta_{obj} + \beta_i \sin\theta_{obj} \tag{2.61}$$

where l_i is the distance from the contact vertex Q to the center of mass, θ_{obj} is the current object angle, and $\Delta\theta_i$ is an offset parameter. We primarily use the form described in equation (2.61), which is linear w/respect to the lumped parameters α_i, β_i.

Since the object is in point/line contact with the ground, the net resultant torque, τ_Q that the hand and gravity exert about the environment contact, Q is zero:

$$\tau_Q = \tau + (\vec{r}_C - \vec{r}_Q) \times \vec{F}_h + \alpha_i \cos\theta_{obj} + \beta_i \sin\theta_{obj}, \qquad \tau_Q = 0 \tag{2.62}$$

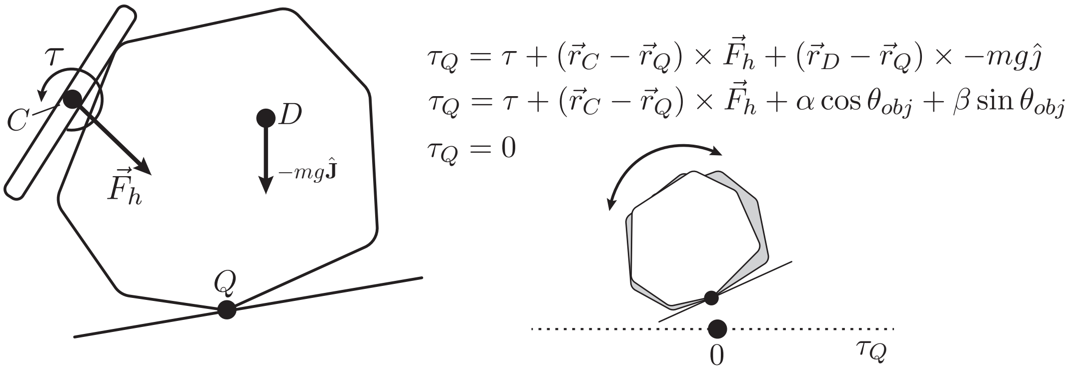

τ
C
D
$\vec{F}_h$
$-mg\hat{\mathbf{J}}$
Q
$\tau_Q = \tau + (\vec{r}_C - \vec{r}_Q) \times \vec{F}_h + (\vec{r}_D - \vec{r}_Q) \times -mg\hat{\jmath}$
$\tau_Q = \tau + (\vec{r}_C - \vec{r}_Q) \times \vec{F}_h + \alpha \cos\theta_{obj} + \beta \sin\theta_{obj}$
$\tau_Q = 0$
0
τ_Q

Figure 2-19: The torque constraints for the object/environment interface during point contact. **Left:** Partial free body diagram of the system (environment wrench not pictured). During point contact, we take into account the effect of gravity on the system. We express the gravitational moment about the environment contact, Q, as a function of the object's orientation θ_{obj}, and the parameters α, β (which depend on the contact vertex). **Right:** The net resultant torque τ_Q that the hand and gravity exert about the ground contact Q is zero. In other words, during point contact, Q is the COP of the ground contact patch.

2.3.3 Basic Contact Constraints

Just as a quick note: we sometimes make use of the wrench constraints that enforce the basic contact conditions (Figure 2-20). These correspond to the following inequalities on the end-effector wrench:

$$F_{hn} \geq 0 \quad \text{(hand contact)} \tag{2.63}$$

$$mg - F_{hy} \geq 0 \quad \text{(ground contact)} \tag{2.64}$$

Since we do not estimate the mass of the the object directly, we use a stricter inequality for the ground contact (that assumes a massless object):

$$-F_{hy} \geq 0 \tag{2.65}$$

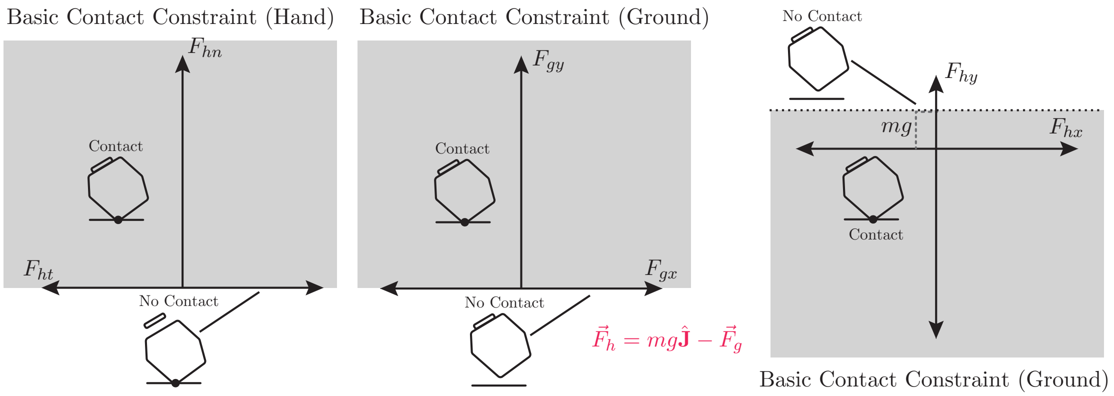

Basic Contact Constraint (Hand)
F_{hn}
Contact
F_{ht}
No Contact
Basic Contact Constraint (Ground)
F_{gy}
Contact
F_{gx}
No Contact
$\vec{F}_h = mg\hat{\mathbf{J}} - \vec{F}_g$
No Contact
F_{hy}
mg
F_{hx}
Contact
Basic Contact Constraint (Ground)

Figure 2-20: Basic wrench constraints used to enforce contact at both the hand and ground interfaces.

2.3.4 Kinematic Constraints and Admissible Motion Directions

Now that we have figured out all of the wrench-space constraints for our system, it is time to enumerate the kinematic constraints and admissible motion directions associated with the various discrete contact modes that are feasible in our system. We will express the admissible motion directions in terms of end-effector pose displacements, which is the form used by our controller. The framing of the kinematic constraints is a little less consistent, and varies on a case-to-case basis. We continue to make use of the system parameters described in Section 2.1.3.

We'll start by looking at the constraints/motions associated with the object/ground contact interface (Figure 2-21). We'll consider the cases of sticking/sliding contact and point/line vs. flush contact.

Contact: Regardless of the mode or geometry, contact enforces the constraint that the contacting vertex/vertices of the object must coincide with the ground:

$$\vec{v}_{c,world} \cdot \hat{e}_Y = h_{grnd} \quad \text{(point/line contact)} \tag{2.66}$$

$$\vec{v}_{c0,world} \cdot \hat{e}_Y = h_{grnd}, \quad \vec{v}_{c1,world} \cdot \hat{e}_Y = h_{grnd} \quad \text{(flush contact)} \tag{2.67}$$

where $\vec{v}_{c,world}$ and $(\vec{v}_{c0,world}, \vec{v}_{c1,world})$ are the world-frame positions of the contact vertex/vertices.

Sticking contact: During sticking contact with the ground, the contact vertices do not move in the horizontal direction:

$$\frac{d}{dt}\left(\vec{v}_{c,world} \cdot \hat{e}_X\right) = 0 \quad \text{(point/line contact)} \tag{2.68}$$

$$\frac{d}{dt}\left(\vec{v}_{c0,world} \cdot \hat{e}_X\right) = 0, \quad \frac{d}{dt}\left(\vec{v}_{c1,world} \cdot \hat{e}_X\right) = 0 \quad \text{(flush contact)} \tag{2.69}$$

Sliding contact: During sliding contact with the ground, the system has a horizontal degree of freedom. The corresponding admissible motion direction is given by:

$$\Delta\vec{q}_X = (\hat{e}_X, 0) \tag{2.70}$$

Flush contact: During flush contact, two vertices are constrained to coincide with the ground instead of one which fixes the orientation of the object:

$$\frac{d}{dt}\theta_o = 0 \tag{2.71}$$

Point/line contact: During point/line contact, the object is free to pivot about the external contact location:

$$\Delta\vec{q}_\theta = (\hat{e}_Z \times (\vec{r}_h - \vec{v}_{c,world}), 1) \tag{2.72}$$

We now consider the same set of contact mode/geometry combinations for the object/hand contact (Figure 2-22).

Contact: Regardless of the mode or geometry, contact enforces the constraint that the contacting vertex/face of the object must coincide with the hand:

$$\left(\vec{v}_{c,world} - \vec{r}_h\right) \cdot \hat{n} = 0 \quad \text{(point/line contact)} \tag{2.73}$$

$$\left(\vec{v}_{c0,world} - \vec{r}_h\right) \cdot \hat{n} = 0, \quad \left(\vec{v}_{c1,world} - \vec{r}_h\right) \cdot \hat{n} = 0 \quad \text{(flush contact)} \tag{2.74}$$

where $\vec{v}_{c,world}$ is the world-frame position of the contact vertex/vertices during point/line contact, and $(\vec{v}_{c0,world}, \vec{v}_{c1,world})$ are the endpoints of the object contact face during flush contact.

Sticking contact: During sticking contact with the hand, the contact vertices do not move in the tangential direction (relative to the hand):

$$\frac{d}{dt}\left(\left(\vec{v}_{c,world} - \vec{r}_h\right) \cdot \hat{t}\right) = 0 \quad \text{(point/line contact)} \tag{2.75}$$

$$\frac{d}{dt}\left(\left(\vec{v}_{c0,world} - \vec{r}_h\right) \cdot \hat{t}\right) = 0, \quad \frac{d}{dt}\left(\left(\vec{v}_{c1,world} - \vec{r}_h\right) \cdot \hat{t}\right) = 0 \quad \text{(flush contact)} \tag{2.76}$$

Sliding contact: During sliding contact with the hand, the hand translate in the tangential direction while the object remains fixed in place. The corresponding admissible motion direction is given by:

$$\Delta\vec{q}_t = (\hat{t}, 0) \tag{2.77}$$

Flush contact: During flush contact, the contact faces of the hand and object are parallel. This means that orientation of the hand and the object are the same up to a constant offset:

$$\theta_o - \phi_c = \theta_h \tag{2.78}$$

where ϕ_c is the angle of the object contact face normal (in the object frame).

Point/line contact: During point/line contact, the hand is free to pivot about the object contact, while the object remains fixed in place.

$$\Delta\vec{q}_{\psi} = (\hat{e}_Z \times (\vec{r}_h - \vec{v}_{c,world}), 1) \tag{2.79}$$

Finally, we look at the special case of wall contact (Figure 2-23). During wall contact, object pivoting and sliding cannot be decoupled: they must either occur simultaneously or not at all. In this case, we can construct the admissible motion direction from the location of the instantaneous center of rotation (ICR) of the object during this motion:

$$\Delta\vec{q}_{\theta} = (\hat{e}_Z \times (\vec{r}_h - \vec{r}_Q), 1) \tag{2.80}$$

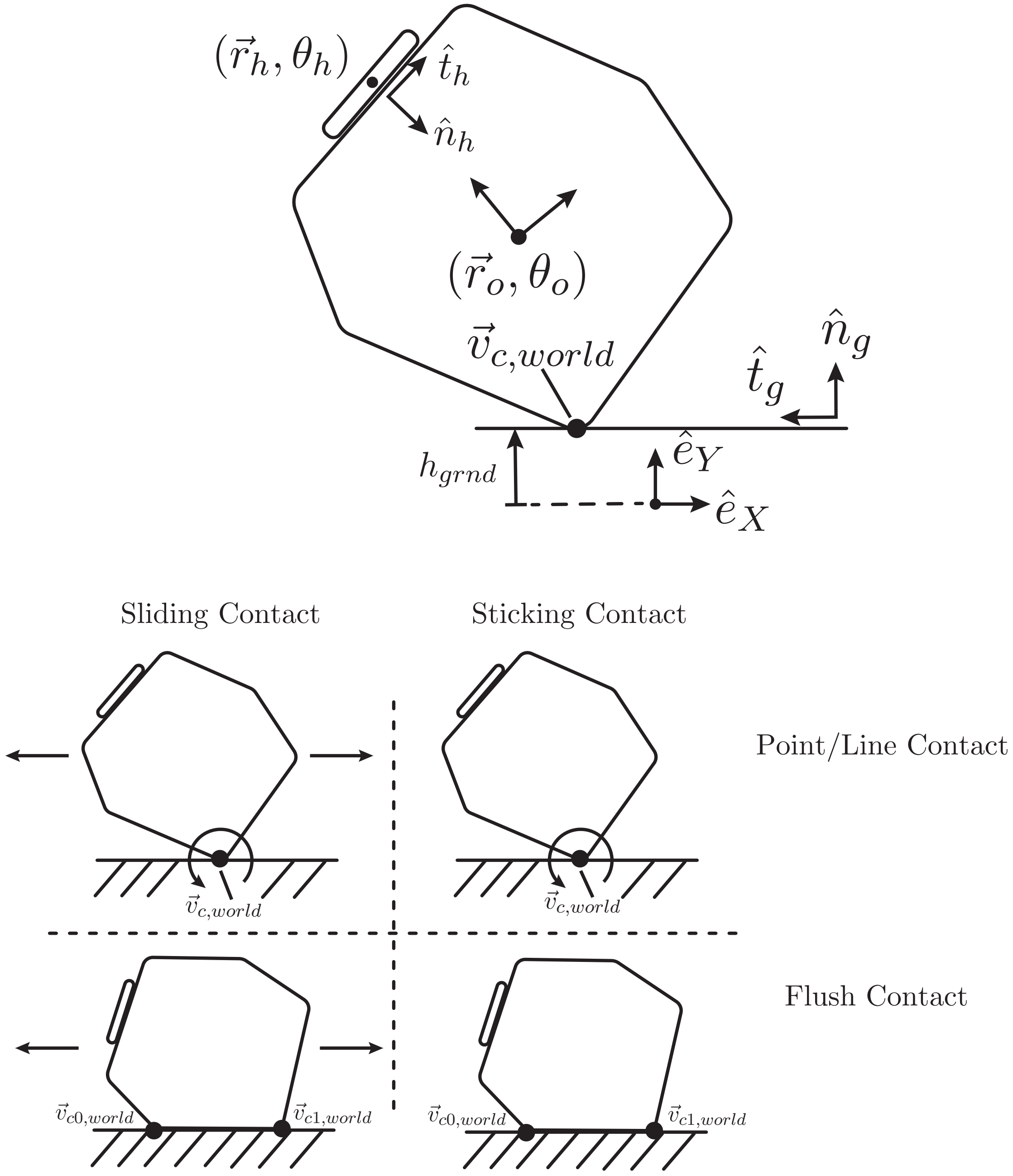


Figure 2-21: Admissible motions associated with the object/ground contact.

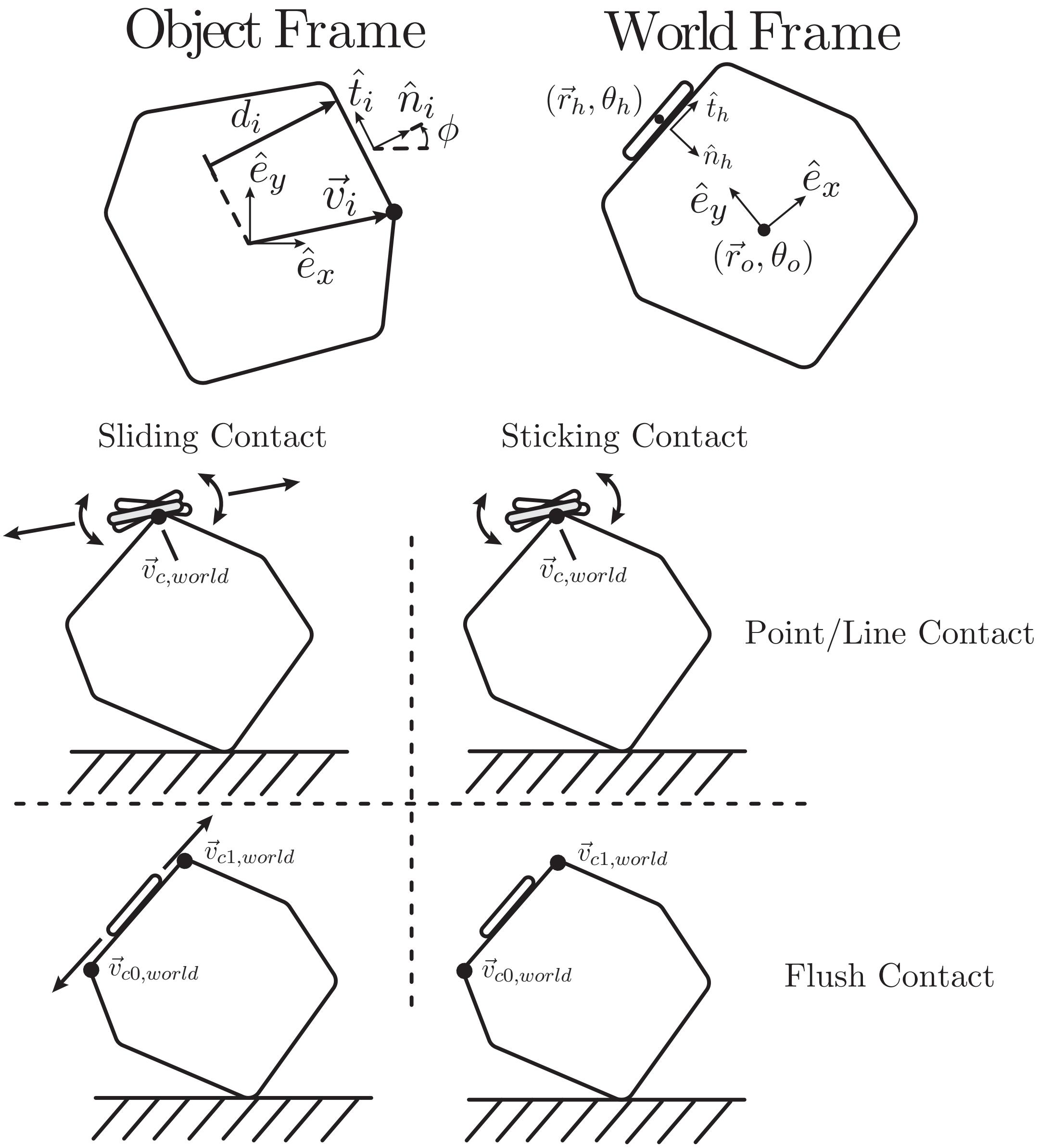


Figure 2-22: Admissible motions associated with the object/hand contact.

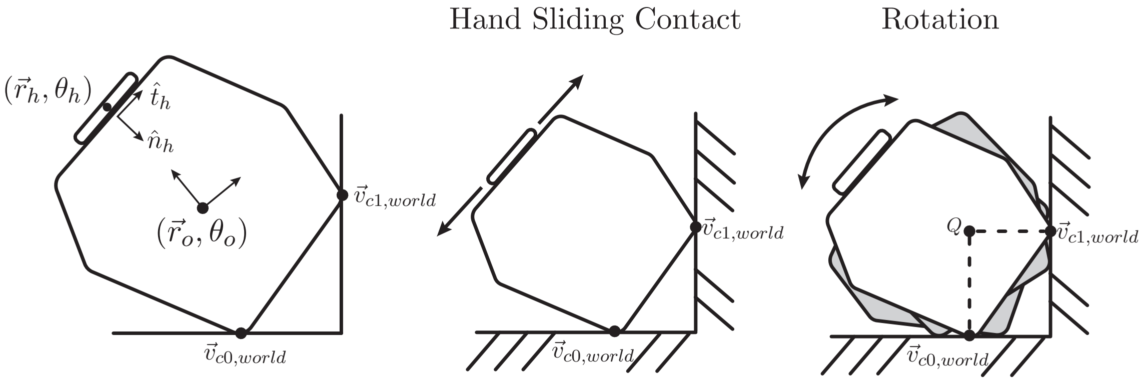


Figure 2-23: Admissible motions associated with wall contact.

Chapter 3

Estimation

The estimator consists of two separate modules: the friction estimator and the kinematic estimator (Figure 3-1). The friction estimator builds a model for the friction cones at the hand and ground contact interfaces, which is used to detect and regulate sticking/sliding at both contacts. The kinematic estimator detects the location and geometry of each contact. This process has two stages. First, we use a set of heuristics to guess the contact geometry of each interface. Second, these estimates (and the stick/slip estimates from the friction module) are mapped to a set of kinematic/wrench constraints, which are then added as constraint factors in a factor graph. The graph is then solved to find an estimate of the system state and parameters. We warm start this loop using a prior provided by vision.

3.1 Friction Cone and Contact Mode Estimation

We want to construct a conservative online estimate of the friction cone boundaries that is robust to sensor noise and possible outliers. A conservative estimate is desirable, as misclassifying a sliding contact as a sticking contact compromises both kinematic estimation and the controller's ability to enforce sticking/flush contact; whereas misclassifying a sticking contact as sliding usually does not cause any problems. The specific issue here is that flipping between sticking and sliding corresponds to activating (sticking) and deactivating (sliding) a kinematic (equality) constraint. The factor graph in the kinematic estimator estimator synthesizes many kinematic constraints to generate an estimate of the object's pose

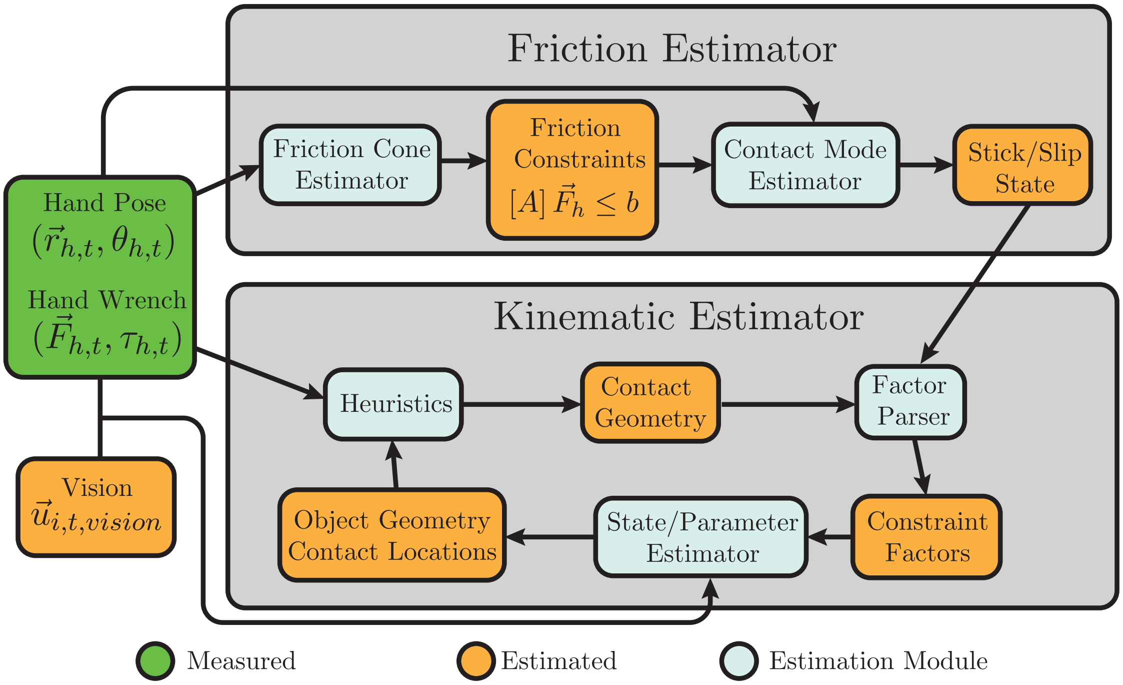

Friction Estimator
Friction Cone Estimator
Friction Constraints $[A]\,\vec{F}_h \leq b$
Contact Mode Estimator
Stick/Slip State
Hand Pose $(\vec{r}_{h,t}, \theta_{h,t})$
Hand Wrench $(\vec{F}_{h,t}, \tau_{h,t})$
Kinematic Estimator
Heuristics
Contact Geometry
Factor Parser
Vision $\vec{u}_{i,t,vision}$
Object Geometry Contact Locations
State/Parameter Estimator
Constraint Factors
Measured
Estimated
Estimation Module

Figure 3-1: The friction and kinematic estimators.

and geometry. This problem is overconstrained, which isn't usually an issue because 1. at least in the ideal case, there is a solution that satisfies all of the constraints, and 2. the problem is framed as an optimization instead of a pure constraint satisfaction problem. As such, removing a single constraint (which would happen when misclassifying sticking as sliding) is usually not too much of an issue, since the other constraints will pick up the slack. However, adding an incorrect constraint (which happens when misclassifying sliding as sticking) that is not consistent with the others is a problem, because it will push the optimization in the wrong direction.

Our estimator relies on the following: (a) that every measured wrench satisfies the wrench constraints, (b) the wrench boundaries are convex, (c) a conservative estimate is sufficient, and (d) the wrench constraints are constant in either the end-effector (2.42-2.43) or world (2.46,2.47) frames.

Hand contact friction constraints: We rewrite the friction constraints at the hand contact

(2.42-2.43) as follows:

$$\mu_{meas} = |(F_{ht}/(F_{hn} + \varepsilon))| < |F_{ht}/(F_{hn})| \leq \mu_h, \tag{3.1}$$

where μ_{meas} is a lower-bound on μ_h and $\varepsilon > 0$ ensures that μ_{meas} is both conservative and well-defined, even when the hand wrench is close to zero. To estimate μ_h, we compute μ_{meas} for each measured wrench. We use all measured data to estimate a single μ_{meas} (instead of one for each boundary) as the hand contact friction cone is symmetric.

We pass the time history of μ_{meas} through a unidirectional low-pass filter (A.3). By unidirectional, we mean that any inputs that are lower than the current output are discarded, resulting in the output being forced to monotonically increase (we discuss how this works in Appendix A). This gives us a conservative estimate for μ_h that reduces the error caused by any large outlying values of μ_{meas} (due to the filter). Essentially, this process tries to find the smallest cone in the space of forces that 1. has a vertex coincident with the origin, 2. has an axis that is aligned with the normal force direction, and 3. contains all of the measured contact forces except for any outliers (Figure 3-3 top right).

Ground contact friction constraints: Unlike the hand contact friction constraints, the ground friction boundaries (2.46,2.47) cannot be rewritten in a highly structured form, as they depend on the unknown mass of the object. This results in the corresponding friction cone being offset in the world-vertical direction. This offset is troublesome: if we attempt to directly fit a cone onto the measured contact forces, there are several situations for which the estimated constraint region is not entirely contained by the true cone (Figure 3-2). This can lead to estimating that a contact is sticking when it is actually sliding, which (as previously mentioned) can comprise both the controller and the kinematic estimator.

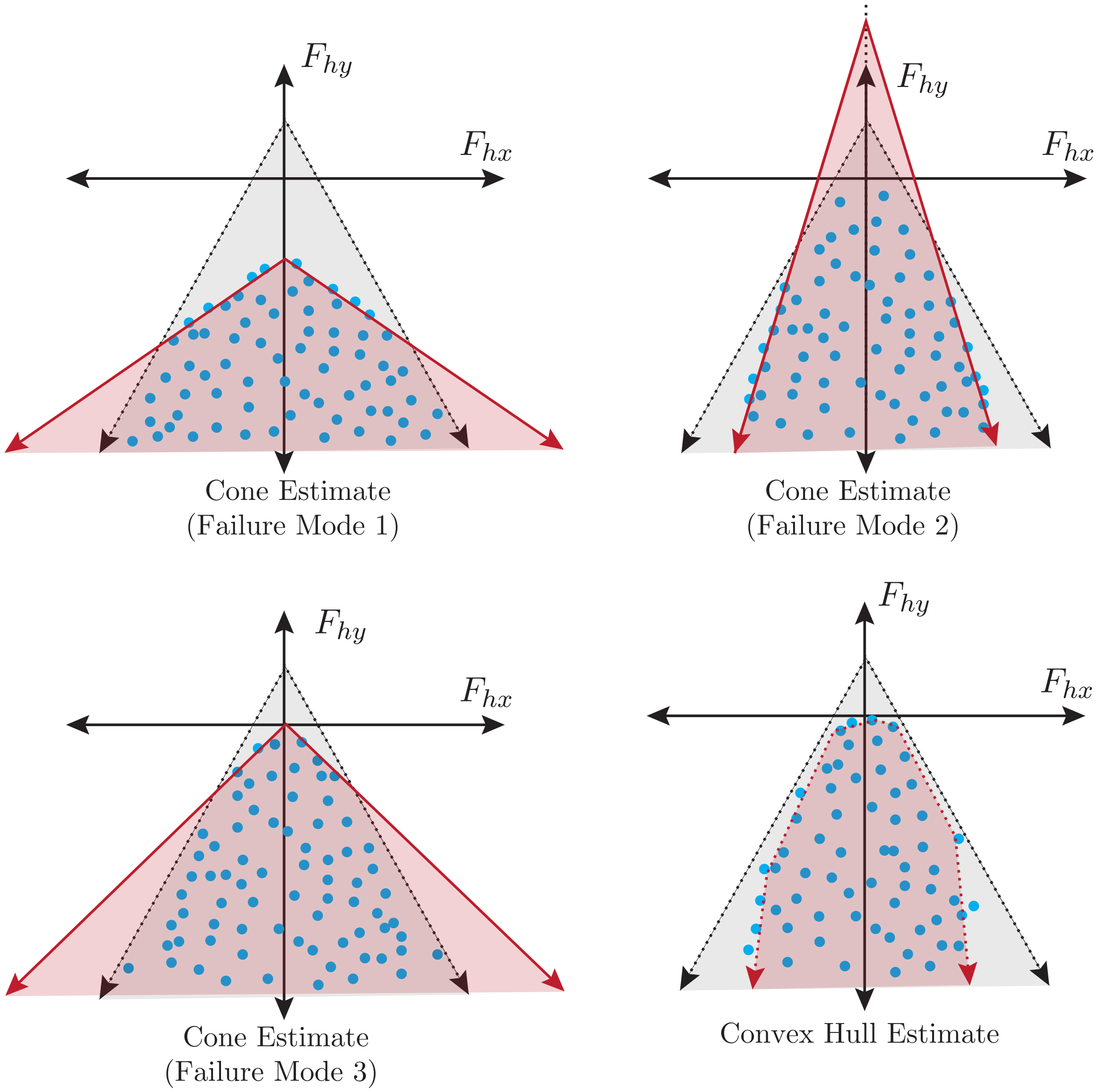
Instead of trying to fit some kind of cone to the data, we directly estimate a convex hull of the force measurements (f_{hx}, f_{hy}) as a conservative approximation of the ground contact friction constraints (Figure 3-3, right). If there is no measurement error, all measured forces will be on the interior of the friction cone for the ground contact, which means that their convex hull will also be entirely contained in this cone (Figure 3-2 bottom right), which is the property that we are looking for. This estimation method is also robust to different ground inclinations (which rotates the friction cone). We compute an online estimate of the convex hull of these forces using the time-history of wrench measurements. This uses the same unidirectional low-pass filter as before to infer a set of supporting hyperplanes of the convex hull. The intersections of these hyperplanes are used to identify candidate corners of the convex hull, which are then refined into the final approximation of the ground contact friction cone. Our method of convex hull estimation is robust to outliers, and favors polygonal estimates with a low number of sides (which reduces the number of linear constraints being generated). A detailed explanation of this estimation process (and why it has these properties) can be found in Appendix A.

Once the convex hull has been estimated, the linear constraints are split (using their boundary normal) into left-sticking constraints (which need to be enforced to prevent the object from sliding to the left), and right-sticking constraints (which are enforced to keep the object from sliding to the right). These constraints and the hand friction cone estimate are then handed to the contact configuration controller (Section 4.2).

Contact mode estimation: Once the friction cones have been estimated, contact mode estimation is pretty straightforward. We predict that the hand is sliding if the measured force is near (or past) the estimated boundaries of the hand contact friction cone, and is sticking otherwise. Similarly, we predict that the object is sliding along the ground if the measured force is near or past the boundaries of the convex hull estimate of the ground contact friction cone, and is sticking otherwise. As previously discussed, this estimation scheme is conservative: sliding/pivoting contact is rarely misclassified as sticking/flush contact, but the reverse misclassification can be frequent. This is intentional as undetected sliding/pivoting introduces error into the kinematic estimator and can result in the system moving into an unrecoverable state.

3.2 Contact Geometry Estimation

We rely on a sequence of heuristics to identify both the active vertices and contact geometry using tactile feedback and the previous state/parameter estimate. These heuristics are built from the kinematic/wrench constraints laid out in Section 2.3 and some geometric



Cone Estimate
(Failure Mode 1)

Cone Estimate
(Failure Mode 2)

Cone Estimate
(Failure Mode 3)

Convex Hull Estimate

Figure 3-2: Various failure modes that can occur when trying to fit a cone to data. **Top row:** If we try to estimate both the vertex of the cone and the slope of the boundaries, it is possible to either end up with either a cone that is too “fat" or an estimated cone whose vertex lies on the exterior of the true cone. Both of these failure modes result in the true cone note entirely containing the estimated cone, which is something that we want to avoid. **Bottom left:** If we fix the vertex of the estimated cone to the origin, it is still possible to end up with an estimate that is too wide. **Bottom right:** Estimating the convex hull sidesteps this issue, since the convex hull is guaranteed to be entirely contained by the true cone.

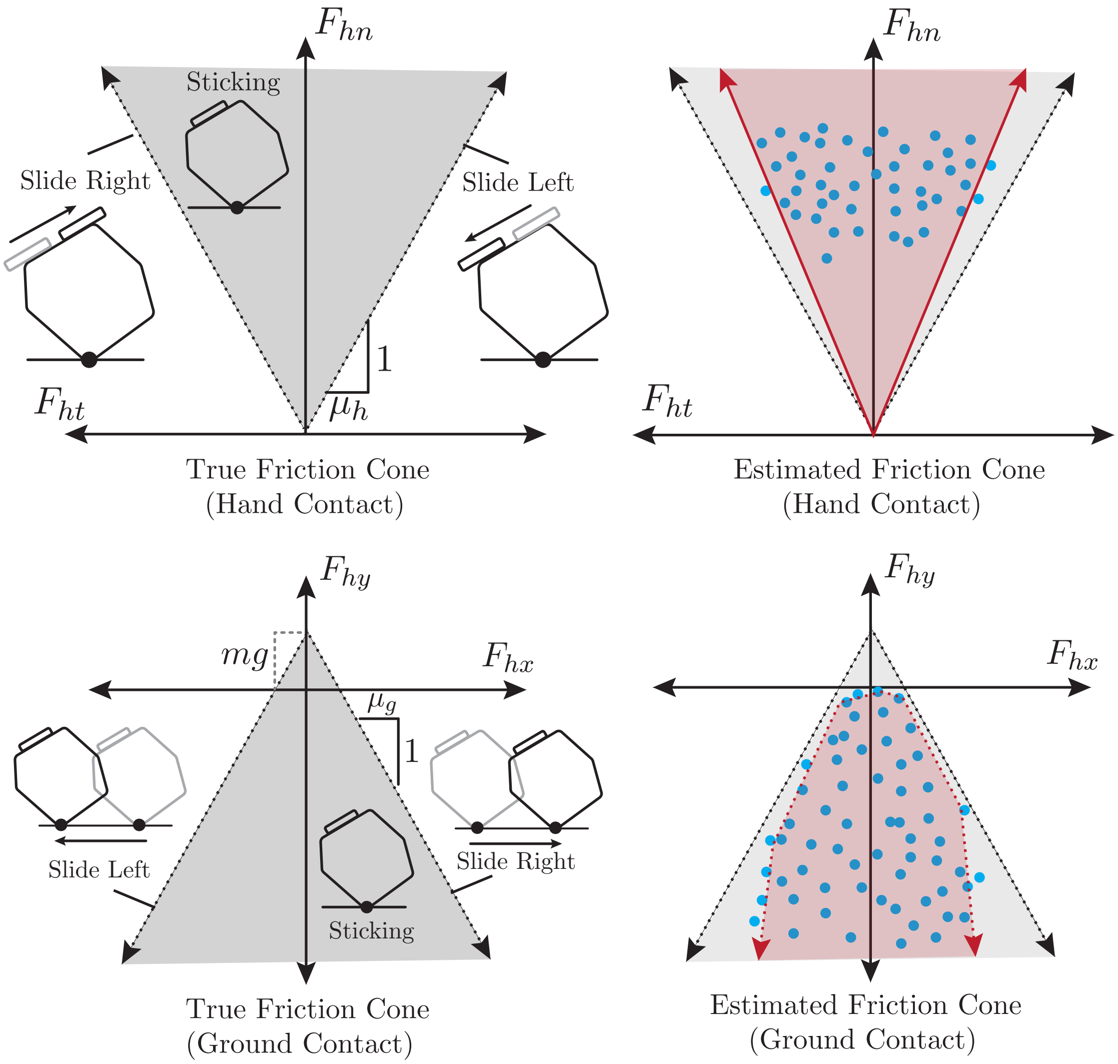


Figure 3-3: **Top left:** The true hand contact friction cone. **Top right:** The estimated hand contact friction cone. **Bottom left:** The true ground contact friction cone. **Bottom right:** The estimated ground contact friction cone.

reasoning.

The object/hand interface: There are four possible contact geometries that we will consider at the object/hand contact interface: no contact, object-line/hand-point, object-point/hand-line and flush contact. We omit object-point/hand-point contact, because it is difficult to maintain and regulate.

Contact detection: The robot and object are assumed to be in contact if the magnitude of the measured force felt at the object/hand contact interface is above a set threshold (around $2-3N$). Below this threshold, we assume no contact.

Object-line/hand-point: To test for this geometry (Figure 2-3b), we use the measured wrench to compute the COP of the hand contact patch. If this COP is sufficiently close to one of the end-effector boundary points, then object-line/hand-point contact is detected.

Object-point/hand-line: For this case (Figure 2-3c), we once again examine the COP of the hand. As per (2.35), during object-point/hand-line contact, the hand COP is coincident with the contact vertex. If the estimated positions of the hand COP and one of the object vertices are close, then object-point/hand-line contact is likely. In this case, we then test the kinematic feasibility of this geometry, taking into account the other active contact constraints. A similar feasibility test is performed for flush contact, and the results are compared to determine which geometry is more likely. We supplement this with a heuristic that leverages intuition of how the hand COP changes as a function of the end-effector pose. During flush contact, the COP often fluctuates while the hand remains stationary; whereas during object-point/hand-line contact, the COP motion is constrained by (2.35). We use this to compare the likelihood of these two geometries.

Flush contact: If contact has been detected, but the previous two geometries have been ruled out as possibilities, then we assume that the system is in flush contact (Figure 2-3a). We then identify the current contact face by comparing the surface normal of the end-effector to the outward facing surface normals of the object in the most recent state estimate.

The object/environment contact(s): Here, we assume that the object is always in contact with the ground. Ground contact is limited to either object-point/ground-line or flush contact. We also check to see if wall contact is occurring.

Object-point/ground-line vs. flush contact: Given the most recent kinematic estimate and tactile feedback, we perform a sequence of tests. First, if one of the object vertices is sufficiently below the rest, then we assume single point contact with the ground at that vertex (Figure 2-3d). Otherwise, if the two lowest object vertices have similar altitude e.g., the bottom edge of the object polygon is near horizontal, we compute the COP of that edge (assuming a weightless object). If the COP is in the interior of this edge by some margin, then we assume that the edge is in flush contact with the ground (Figure 2-3e). Otherwise, the closest vertex to the COP is in single point contact with the ground. Figure 3-4 depicts these three possible cases.

Wall contact: To determine whether or not the object is in contact with a wall (Figure 2-3f), we compare the measured force at the hand contact with the friction constraints that we estimate for the ground contact. These estimated constraints are computed during periods when there are no external contacts besides the ground; after which they are frozen in place (during periods when wall contact is allowed). Wall contact is detected when the measured contact force significantly violates one of the friction constraints for ground contact, since it implies the existence of a new contact that is generating the wrench causing the constraint violation.

3.3 Object and Contact Localization

We now synthesize the contact mode/geometry estimates with tactile feedback to estimate the contact locations.

Factor graphs and GTSAM: We use factor graphs to formulate the problem of localizing the object and contacts (Tactile SLAM [65]). Factor graphs are bipartite graphs which encode variables and constraints as vertices, and their input-output relationships as edges. For our purposes, factor graphs provide a straightforward way to leverage our knowledge of the contact constraints in order to localize the object and contacts (Figure 3-5). Here, the contact geometry factors (depicted in blue) constrain the instantaneous contact pose and wrench, while the contact mode factors (purple) constrain how the contact pose evolves over time. Each factor is expressed as a function that maps the estimated variables to a

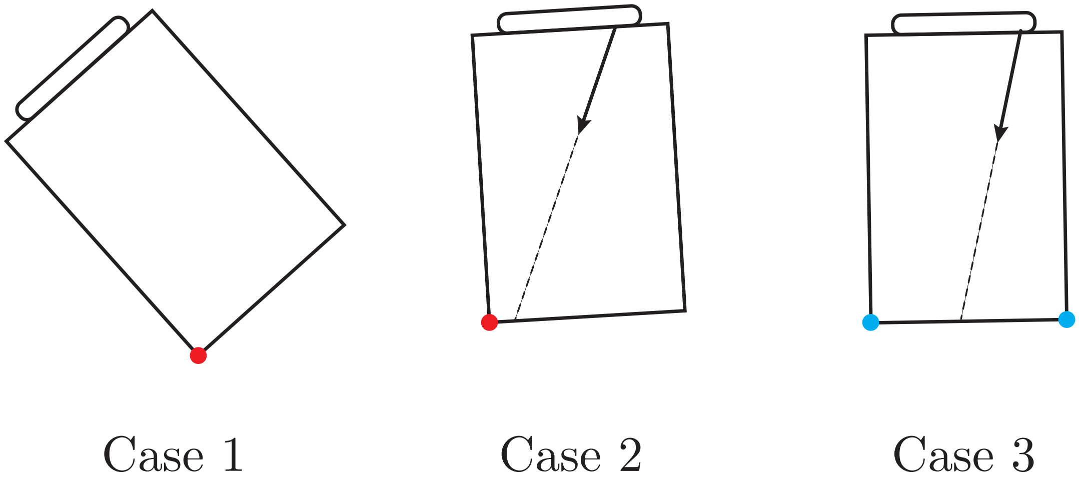


Figure 3-4: Heuristic used to disambiguate between object-point/ground-line and flush contact. **Case 1:** If one vertex is significantly lower than the rest, then it is labelled as the contact vertex. **Case 2:** If there is no "lowest vertex", but the COP is very close to a vertex, then that vertex is labelled as the contact vertex **Case 3:** If no single contact vertex is identified then flush contact is detected, and the two lowest vertices are labelled as being in contact.

constraint violation error. Each constraint has an associated variance that reflects our confidence in its accuracy. There are various algorithms available for computing the optimal values of the estimated variables. We rely on the factor graph software GTSAM [11], and its implementation of the incremental smoothing and mapping algorithm [35]. We provide a brief summary of how factor graphs work in Appendix C.

Variable factors and measured quantities: Using the system parameterization described in Section 2.1.3, the following are variables in our factor graph:

- The object-frame vertex positions, $\vec{v}_i = (x_i, y_i)$.
- The parameters describing the object-frame surface normals ϕ_i and face offsets d_i.
- The object pose in the world-frame, $(\vec{r}_{o,t}, \theta_{o,t})$.
- During flush contact, the tangential displacement between the hand and object, s_t.
- The height of the ground h_{grnd}.
- The gravitational torque parameters (α_i, β_i).

We use F/T sensing and robot proprioception to measure:

- The hand contact wrench $\vec{w}_{h,t} = (\vec{F}_{h,t}, \tau_{h,t})$.
- The hand pose $(\vec{r}_{h,t}, \theta_{h,t})$.

We also assume that the ground is horizontal, meaning that $\hat{n}_g$ and $\hat{t}_g$ are respectively vertical and horizontal.

When describing the constraint factors in this paper, we let $\vec{u}_{i,t}$ denote the current world-frame positions of the object vertices. These are computed as a function of the estimation variables, specifically the object pose and the object-frame vertex positions:

$$\vec{u}_{i,t} = R(\theta_{o,t})\vec{v}_i + \vec{r}_{o,t} \tag{3.2}$$

where $R(\theta)$ is the rotation matrix for angle θ. When the hand is in flush contact with object face c, we use:

$$\vec{u}_{i,t} = R(\theta_{h,t} - \phi_c + \pi)\vec{v}_i + d_c\hat{n}_{h,t} + s_t\hat{t}_{h,t} \tag{3.3}$$

The estimator may be provided with noisy live visual feedback of the world-frame vertex positions, $\vec{u}_{i,t,vision}$.

Constraint factors: We convert the estimated contact mode and geometry into a set of constraint factors that are added to the factor graph at each time-step.

Contact: The end-effector intersects any object point(s) it is in contact with. The same holds true for the ground:

$$\left(\vec{u}_{i,t} - \vec{r}_{h,t}\right) \cdot \hat{n}_{h,t} = 0 | i \in \{\text{hand contact vertices}\} \tag{3.4}$$

$$\vec{u}_{i,t} \cdot \hat{n}_g - h_{grnd} = 0 | i \in \{\text{ground contact vertices}\} \tag{3.5}$$

During flush contact with the hand, the hand contact constraint is implicit to (3.3), so we do not include a factor.

Sticking contact: If the object is in sticking contact with the hand, then the tangential motion (relative to the hand) of the contact point(s) is zero:

$$\left(\vec{u}_{i,t+1} - \vec{r}_{h,t+1}\right) \cdot \hat{t}_{h,t+1} - \left(\vec{u}_{i,t} - \vec{r}_{h,t}\right) \cdot \hat{t}_{h,t} = 0 \tag{3.6}$$

$$s_{t+1} - s_t = 0 \;\; \text{(flush contact)} \tag{3.7}$$

Similarly, if the object is in sticking contact with the ground, the horizontal motion of the ground contact(s) is zero:

$$(\vec{u}_{i,t+1} - \vec{u}_{i,t}) \cdot \hat{t}_g = 0 \tag{3.8}$$

Torque Balance: When the object is in single point contact with ground, but not in wall contact, then by the quasi-static assumption, the net torque about the ground contact is zero.

$$\left(\vec{r}_{h,t} - \vec{u}_{i,t}\right) \times \vec{F}_{h,t} + \tau_{h,t} + \alpha_i \cos\theta_{o,t} + \beta_i \sin\theta_{o,t} = 0 \tag{3.9}$$

Here, the quantity $\alpha_i \cos\theta_{o,t} + \beta_i \sin\theta_{o,t}$ represents the gravitational torque about the ground contact vertex.

When the object is in single point contact with the hand, then the object contact vertex should be coincident with the COP of the hand contact patch, which we can compute by applying (2.37) to the hand pose and measured contact wrench:

$$\vec{u}_{i,t} - \vec{r}_{COP,h,t} = 0, \;\; i = \text{hand contact vertex} \tag{3.10}$$

Vision: Any vision estimates are added as extra constraints:

$$\vec{u}_{i,t} - \vec{u}_{i,t,vision} = 0 \tag{3.11}$$

Geometric consistency: These constraints ensure that object surface normal ϕ_i and face offset d_i parameters are geometrically consistent with the object vertices $\vec{v}_i$:

$$d_i = \hat{n}_i \cdot \vec{v}_i \;\rightarrow\; x_i \cos\phi_i + y_i \sin\phi_i - d_i = 0 \tag{3.12}$$

$$d_i = \hat{n}_i \cdot \vec{v}_{i+1} \;\rightarrow\; x_{i+1} \cos\phi_i + y_{i+1} \sin\phi_i - d_i = 0 \tag{3.13}$$

Regularization constraints: We include low-weight constraint factors to limit the object motion between time-steps.

$$\vec{r}_{o,t+1} - \vec{r}_o = 0, \;\; \theta_{o,t+1} - \theta_{o,t} = 0 \tag{3.14}$$

$$s_{t+1} - s_t = 0 \;\; \text{(flush contact)} \tag{3.15}$$

Optimization weights: Each constraint factor has an associated variance that reflects our confidence in its accuracy. This is essentially an optimization weight, with lower variances increasing the significance of their corresponding factor in the optimization. We tuned these variances manually.

3.4 Estimator Initialization and the Role of Vision

The contact geometry heuristics (Section 3.2) and the factor graph estimator (Section 3.3) are mutually dependent: the contact geometry estimator needs a somewhat accurate state/parameter estimate to disambiguate contact geometries through tactile feedback, and the factor graph based estimator relies on the contact geometry estimator to determine which constraints are imposed at each time-step. This presents a chicken and egg problem: if the initial estimation error is too large, it will never correctly converge.

We resolve this by using vision to generate a prior of the object's geometry and starting pose to warm start the kinematic estimator. Once it has been initialized, the kinematic estimator is capable of operating on tactile feedback alone. Live visual feedback is still useful when available, and acts to regularize the object pose. This safeguards against errors from accumulating when the heuristics incorrectly guess the contact geometries.

The vision heuristic we use directly estimates the positions of the object vertices in

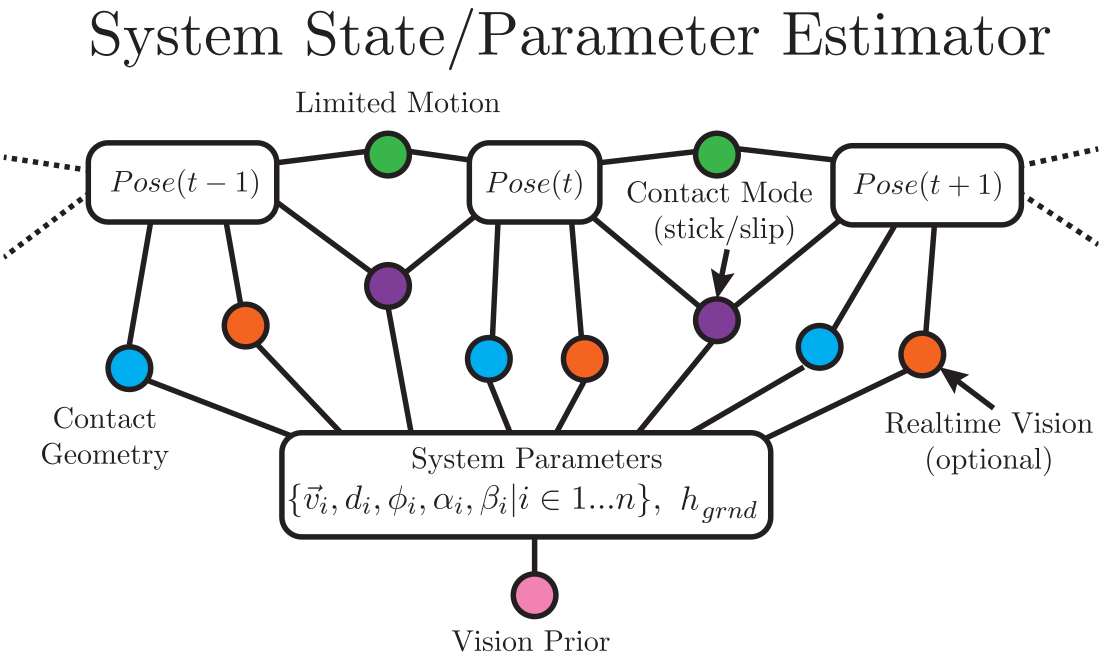

System State/Parameter Estimator
Limited Motion
$Pose(t-1)$
$Pose(t)$
Contact Mode (stick/slip)
$Pose(t+1)$
Contact Geometry
Realtime Vision (optional)
System Parameters
$\{\vec{v}_i, d_i, \phi_i, \alpha_i, \beta_i | i \in 1 \ldots n\},\ h_{grnd}$
Vision Prior

Figure 3-5: Factor graph used for the kinematic estimator. Different sets of contact mode and geometry factors are used at each time-step, with the composition depending upon the stick/slip and contact geometry estimates.

the world-frame, under the assumption that the object is a convex polygon. We discuss our method in detail in Appendix B. This heuristic was sufficiently reliable for developing the kinematic estimator, but should be replaced with a standard perception method during real-world implementation.

Chapter 4

Control

The contact configuration controller (Figure 4-1) regulates both the pose of the object and end-effector, as well as the mode and geometry at the various contact interfaces of the system. To determine which motions to execute, the controller listens for motion commands (currently generated by a human, though ideally generated by a high-level planner). Each command comes in the form of a motion primitive, which specifies the desired motion as well as the mode and geometry of the contacts. The contact configuration controller then parses this motion primitive, converting it into a quadratic program (QP), whose solution is the pose of an impedance target. This target pose is then handed to a low-level impedance controller that executes the motion. The controller is supported by the estimator, which provides the information about the geometry and friction properties of the system that is used to generate the QP.

4.1 Impedance Control

Our contact configuration controller is built on top of a low-level Cartesian impedance controller. In impedance control, the controller emulates the behavior of a mass-spring-damper system. For our case, this behavior is realized through the following control law:

$$\vec{w}_h = -\mathbf{K}(\vec{x}_h - \vec{x}_{tar}) - \mathbf{B}\frac{d\vec{x}_h}{dt} \tag{4.1}$$

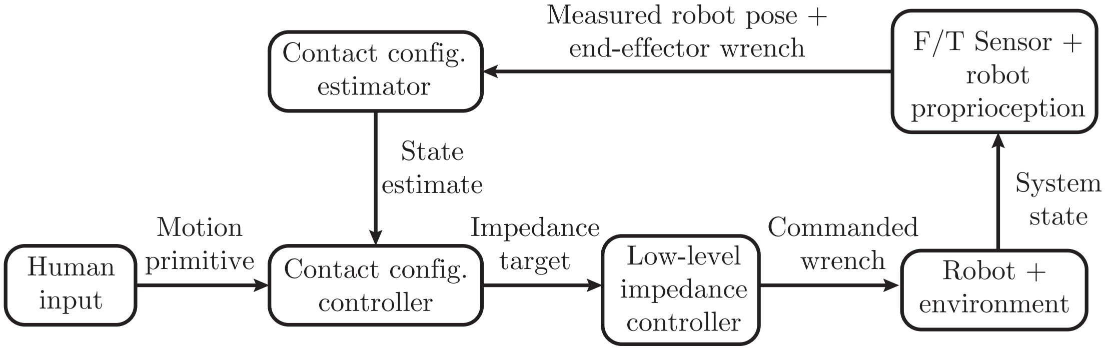


Figure 4-1: The contact configuration controller converts motion primitives (provided by a human) into a quadratic program. The solution to this quadratic program is the pose of an impedance target, which is then passed down to a low-level impedance controller that executes the motion. The controller is aided by the estimator, which provides the necessary information to construct the QP.

where $\vec{w}_h = (\vec{F}_h, \tau)$ and $\vec{x}_h = (\vec{r}_h, \theta_h)$ are the end-effector wrench and pose respectively, $\vec{x}_{tar} = (\vec{r}_{tar}, \theta_{tar})$ is the pose of a virtual impedance target, and $(\mathbf{K}, \mathbf{B})$ are the stiffness and damping matrices (which are input parameters to the controller).

Under the quasi-static assumption, the damping term can be ignored because the system is assumed to always be at rest:

$$\vec{w}_h = -\mathbf{K}(\vec{x}_h - \vec{x}_{tar}) \tag{4.2}$$

Figure 4-2 illustrates this virtual spring model for the impedance controller. Our contact configuration controller regulates both $\vec{x}_h$ and $\vec{w}_h$ by updating $\vec{x}_{tar}$ while keeping $\mathbf{K}$ constant. Using an impedance target as a proxy for regulating the end-effector pose is called indirect force control (IFC). IFC provides a natural interface for simultaneously regulating both the end-effector pose and wrench, which is useful for our contact configuration controller.

Even though damping is not considered when developing our contact configuration controller, it is still very important. Specifically, we rely on damping in order to enforce the quasi-static assumption. Up to a point, increasing the virtual damping reduces the time it takes for the system to return to rest after it has been perturbed (either due to an external wrench or a change in the impedance target pose). Unfortunately, since this virtual damping is being emulated on a computer (which introduces a latency), there is a fundamental limit

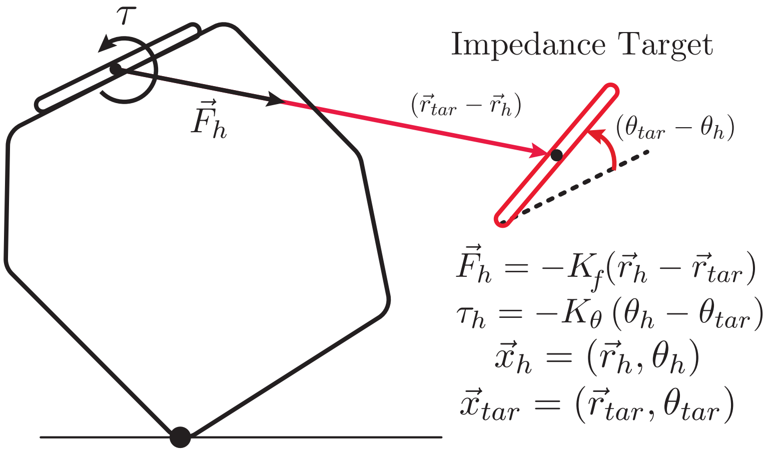

τ
Impedance Target
$(\vec{r}_{tar} - \vec{r}_h)$
$\vec{F}_h$
$(\theta_{tar} - \theta_h)$
$\vec{F}_h = -K_f(\vec{r}_h - \vec{r}_{tar})$
$\tau_h = -K_\theta\,(\theta_h - \theta_{tar})$
$\vec{x}_h = (\vec{r}_h, \theta_h)$
$\vec{x}_{tar} = (\vec{r}_{tar}, \theta_{tar})$

Figure 4-2: An illustration of the virtual spring model for impedance control. Here, the wrench exerted by the end-effector is proportional to the pose error, taken w/respect to an impedance target.

to how large the damping can be before the impedance controller becomes unstable. In our implementation, we have tuned the damping via trial and error to approximate quasi-static behavior as much as possible (without the system becoming unstable).

We have found that a larger stiffness is preferable in practice. A controller with a low stiffness requires a much larger displacement of the impedance target in order to generate the same wrench as a controller with a high stiffness (Figure 4-3). When the impedance target is far away from the true end-effector pose, it is more likely for the system to become statically unstable. The impedance controller is trying to move the end-effector towards the target pose. When this target pose is far away, a small perturbation from the current equilibrium can cause the system to pick up momentum as the end-effector is brought towards this “natural" equilibrium state: this is the root of the static instability. When the stiffness is large and the impedance target is nearby, the motion of the end-effector is far more limited, which prevents such instabilities from occurring.

In addition, a high stiffness mitigates the consequences of the Stribeck effect (i.e. the friction coefficient decreasing during slipping). Because the dynamic friction coefficient is lower than the static friction coefficient, during the transition from sticking to slipping it is easy to overshoot the target state before the system returns to rest. The amount of potential overshoot is reduced as the impedance target gets closer to the true end-effector pose. Thus, a larger stiffness is better.

Similar to damping, the latency introduced by the computer places a fundamental limitation on how large the stiffness can be before the system becomes unstable. As such, we have also tuned the stiffness via trial and error to be as large as possible (while keeping the system stable).

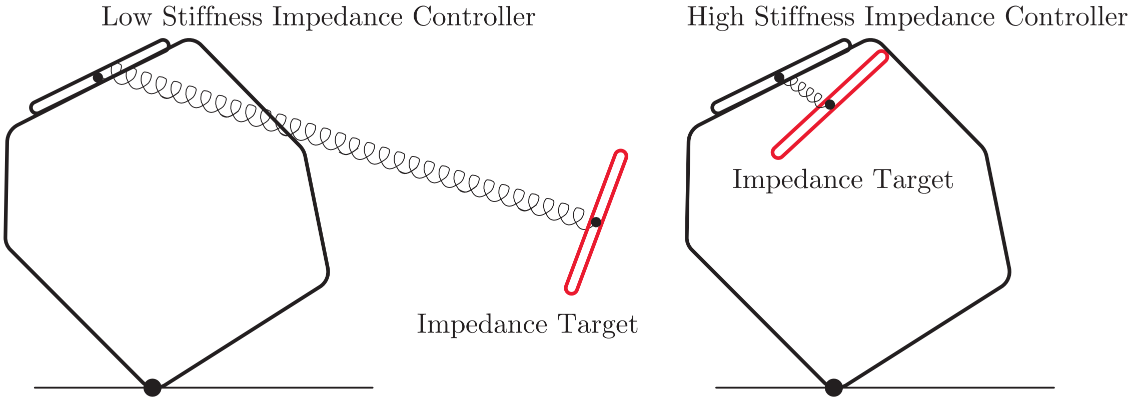

Low Stiffness Impedance Controller
Impedance Target
High Stiffness Impedance Controller
Impedance Target

Figure 4-3: A low stiffness impedance controller requires a much larger pose error (w/respect to the impedance target) to generate the same wrench as a high stiffness impedance controller.

4.2 Contact Configuration Controller

The contact configuration controller regulates the system's pose along the admissible motion directions (Section 2.3.4), while simultaneously regulating the hand contact wrench to maintain the desired contact mode/geometry. This simultaneous regulation of both forces and motions is achieved through the aforementioned impedance control layer (IFC).

We rely on the intuition that, for a fixed contact mode/geometry, moving the impedance target parallel to the admissible motion directions will generate motions along those directions (Figure 4-4); and moving the target in a direction that is normal to the active kinematic constraints will generate reaction wrenches in those directions to enforce said constraints, allowing for wrench regulation (Figure 4-5).

Given a set of desired contact modes/geometries and system motions (i.e. a motion

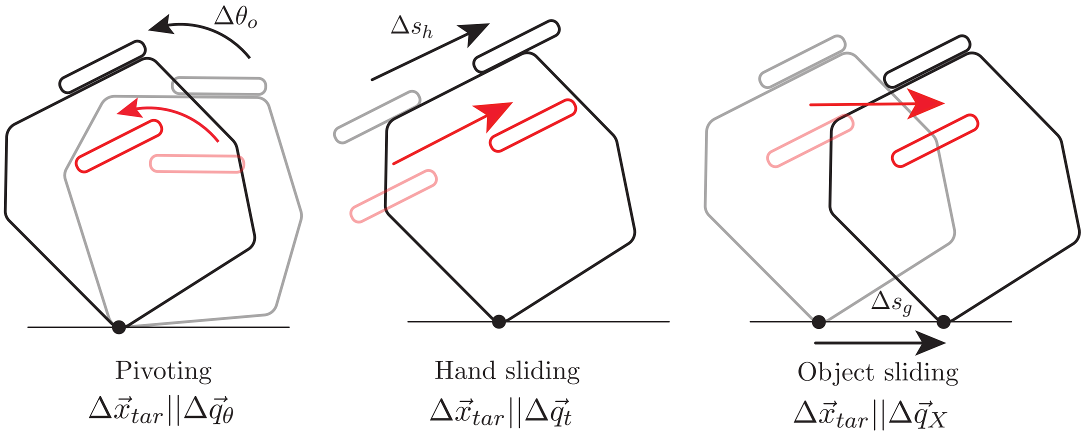

$\Delta\theta_o$
Δs_h
Δs_g
Pivoting
$\Delta\vec{x}_{tar}||\Delta\vec{q}_\theta$
Hand sliding
$\Delta\vec{x}_{tar}||\Delta\vec{q}_t$
Object sliding
$\Delta\vec{x}_{tar}||\Delta\vec{q}_X$

Figure 4-4: Moving the impedance target along the admissible motion directions will drag the system's pose along with it.

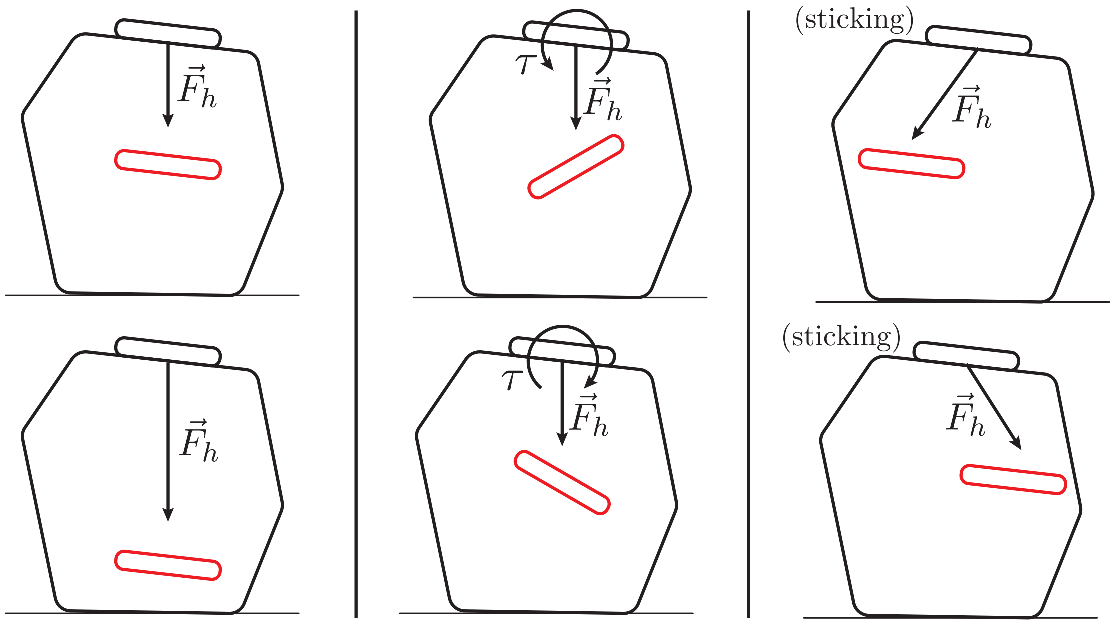


Figure 4-5: Moving the impedance target in a direction that is normal to the active kinematic constraints allows for wrench regulation in those directions.

primitive), we need to regulate the impedance target in a way that enforces the necessary wrench constraints while generating the desired motion. To accomplish this, at each timestep the controller solves a quadratic program (QP) to determine the incremental change of the impedance target, $\Delta\vec{x}_{tar}$:

$$\min_{\Delta\vec{w}_h,\Delta\vec{x}_{tar}} \quad \alpha_0||\Delta\vec{x}_{tar}||^2+\sum\alpha_i\big(\Delta\vec{x}_{tar}\cdot\Delta\vec{q}_i-\beta_i\Delta\varepsilon_i\big)^2 \tag{4.3}$$

$$s.t. \qquad \hat{n}_j\cdot(\vec{w}_{h,meas}+\gamma_j\Delta\vec{w}_h)\leq b_j \quad \forall j \tag{4.4}$$

$$\Delta\vec{w}_h=\mathbf{K}\Delta\vec{x}_{tar} \tag{4.5}$$

Here, the sum $\vec{w}_{meas}+\Delta\vec{w}_h$ is the predicted wrench that the hand will exert after the impedance target has been incremented by $\Delta\vec{x}_{tar}$. The constraints (4.4) correspond to the estimated value of the wrench space constraints, which are weighted by γ_j. The equality constraint (4.5) corresponds to the stiffness law of the low-level impedance controller. Finally, the cost (4.3) is used to regulate the pose along the admissible motion directions of the system, $\Delta\vec{q}_i$. The composition of cost terms and wrench constraints is specified by the motion primitive that the controller has been commanded to execute. Let's look a bit more closely at the structure of the costs, wrench constraints, and the stiffness equality to better understand whats going on.

Cost terms: The quadratic cost consists of two parts. The first, $\alpha_0||\Delta\vec{x}_{tar}||^2$ is a regularization term that penalizes for large changes in the impedance target location between time-steps. The second, $\sum\alpha_i\big(\Delta\vec{x}_{tar}\cdot\Delta\vec{q}_i-\beta_i\Delta\varepsilon_i\big)^2$, is a summation that encourages the impedance target to move along the different admissible motion directions, $\Delta\vec{q}_i$. Specifically, to minimize the ith cost term, the impedance target needs to move by $\beta_i\Delta\varepsilon_i$ in the direction of the ith admissible motion direction, where $-\Delta\varepsilon_i$ is the estimated pose error along that direction. The weights α_i and β_i are used to assign different levels of priority to each of the motion direction terms. Table 4.1 lists the different types of admissible motion vectors that are used in the QP.

Inequality constraints: The inequality constraints of the quadratic program are slightly modified versions of the wrench-space inequalities associated with whichever contact modes and geometries are being commanded by the motion primitive (Section 2.3). Here, $\hat{n}_j$ and b_j are the constraint normal and offset parameter for the jth wrench-space boundary. The scaling factors γ_j are used to amplify or attenuate wrench corrections. We set these weights dynamically (changing in between time-steps) depending on whether or not the current measured wrench satisfies or violates the corresponding constraint, with violated constraints receiving larger weights (resulting in a larger correction). Table 4.2 lists the different types of wrench constraints that are used in the QP.

Stiffness law equality: The equality constraint, $\Delta\vec{w}_h = \mathbf{K}\Delta\vec{x}_{tar}$, models the stiffness law of the impedance controller, and is the only part of the QP that relates the end-effector wrench to the impedance target pose. In actuality, the stiffness law implies that:

$$\vec{w}_h = \mathbf{K}(\vec{x}_{tar} - \vec{x}_h) \quad \rightarrow \quad \Delta\vec{w}_h = \mathbf{K}(\Delta\vec{x}_{tar} - \Delta\vec{x}_h) \tag{4.6}$$

where $\Delta\vec{x}_h$ is the resulting increment of the end-effector pose. The equality constraint (4.5) implicitly assumes that $\Delta\vec{x}_h = 0$. Indeed, we are making two very contradictory assumptions in the construction of our QP:

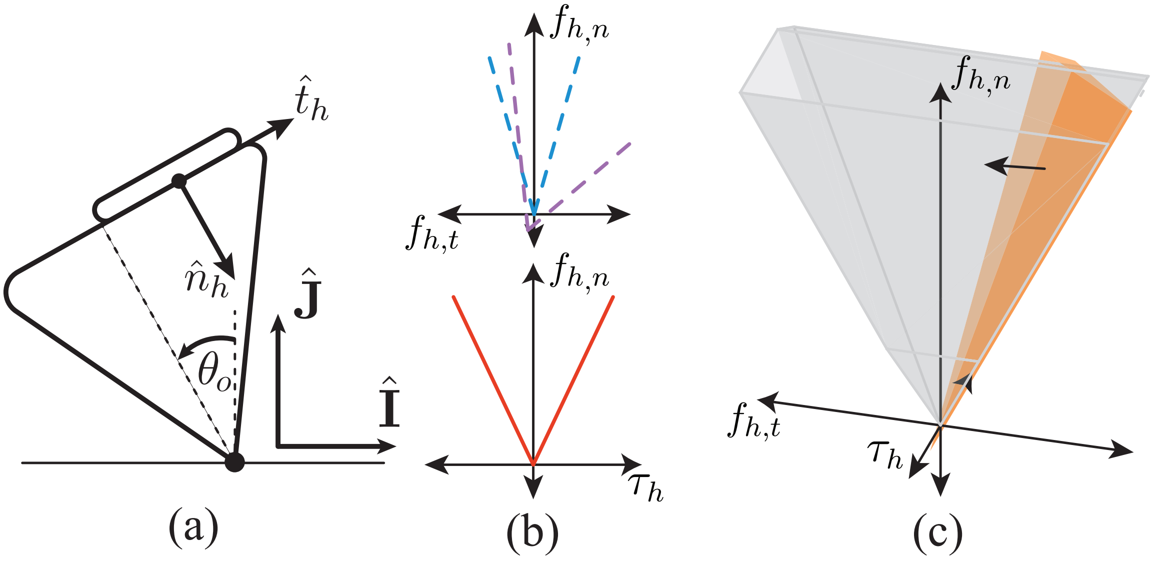
$$\text{Pose regulation (cost):} \quad \Delta\vec{x}_h = \Delta\vec{x}_{tar}, \;\; (\Delta\vec{w}_h = 0) \tag{4.7}$$

$$\text{Wrench regulation (inequalities):} \quad \Delta\vec{w}_h = \mathbf{K}\Delta\vec{x}_{tar}, \;\; (\Delta\vec{x}_h = 0) \tag{4.8}$$

I think the best way to resolve this contradiction is to think about (4.7) and (4.8) as a pair of feedback laws relating errors in the pose and wrench to the desired correction in the impedance target pose, with the QP providing a means to balance between the two.

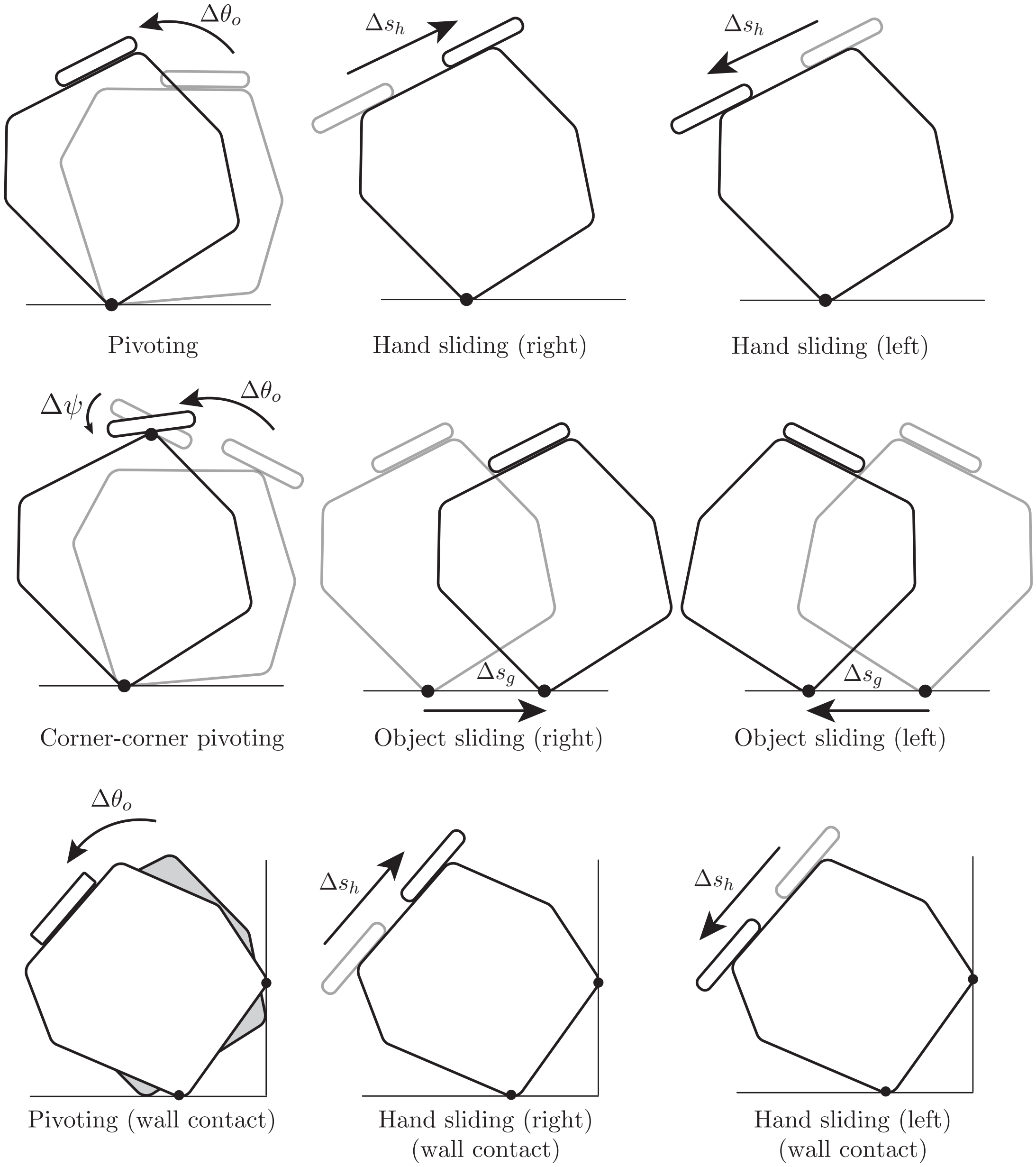
4.3 Motion Primitives

Commands to the controller are provided in the form of motions primitives. Each primitive specifies the contact mode and/or geometry at both the object/hand and object/environment contacts, as well as a desired motion of the system. For instance, the pivot primitive drives the object to rotate, while maintaining flush sticking contact with the hand and sticking contact with the ground (Figure 4-6). We have implemented a variety of primitives (illustrated in Figure 4-7). The contact configuration controller then parses the motion primitive command, assembling the corresponding QP from a standard set of cost terms (listed in Table 4.1) and wrench constraints (listed in Table 4.2). These building blocks (the admissible motion directions and wrench constraints) are provided by the estimator (Chapter 3). For each of the motion primitives depicted in Figure 4-7, we have listed the corresponding set of motion directions and constraints in Table 4.3. One last thing to note is that, though it's not listed, every motion primitive uses a maximum normal force constraint, $F_{hn} \leq f_{max}$, to prevent the robot from crushing the object.



QP costs and constraints

Figure 4-6: The QP cost and constraints for the pivoting primitive. (a) A diagram showing the system state. (b) The corresponding QP constraints for maintaining sticking at both contacts (top) and maintaining line-contact with the hand (bottom). The feasible region of the QP is the intersection of the estimated ground friction cone (dashed purple, top), hand friction cone (dashed blue, top), and prescribed hand torque cone (solid red, bottom). Since these constraints are drawn in the hand frame, the ground friction cone is rotated clockwise by θ_o. (c) The boundary of the QP's 3D feasible region is shown in gray, the target direction associated with regulating the object's orientation, $\Delta\vec{q}_\theta$, is shown in black, and the intersection of the minimum-cost hyperplane defined by $\Delta\vec{q}_\theta$ and $\Delta\theta_o$ with the feasible region shown in orange.



Pivoting

Hand sliding (right)

Hand sliding (left)

Corner-corner pivoting

Object sliding (right)

Object sliding (left)

Pivoting (wall contact)

Hand sliding (right)
(wall contact)

Hand sliding (left)
(wall contact)

Figure 4-7: Various motion primitives that we have implemented in our control framework.

Table 4.1: Admissible motion directions

Admissible Motion Direction	Symbol	Eq.
Object rotation	$\Delta\vec{q}_\theta$	(2.72)
Object translate left	$-\Delta\vec{q}_x$	(2.70)
Object translate right	$+\Delta\vec{q}_x$	(2.70)
Hand translate left	$-\Delta\vec{q}_t$	(2.77)
Hand translate right	$+\Delta\vec{q}_t$	(2.77)
Object rotation (wall)	$\Delta\vec{q}_\theta$	(2.80)
Hand relative rotation	$\Delta\vec{q}_\psi$	(2.79)

Table 4.2: Wrench constraints

Admissible Motion Direction	Description	Eq.
Ground contact	Enforces contact w/ground	(2.65)
Ground sticking (left)	Prevents object sliding left	(2.47)
Ground sticking (right)	Prevents object sliding right	(2.46)
Hand contact	Enforces contact w/hand	(2.63)
Hand sticking (left)	Prevents hand sliding left (relative to obj.)	(2.42)
Hand sticking (right)	Prevents hand sliding right (relative to obj.)	(2.43)
Hand flush-contact	Enforces flush-contact between hand and obj.	(2.50, 2.51)
Environment flush-contact	Enforces flush-contact between obj. and env.	(2.53)
Ground sticking w/environment support	Enforces sticking w/ground but half of friction cone not used b/c of wall	(2.47) or (2.46)
Bounded radial force	bounds force in radial direction	$a \leq f_{rad} \leq b$

Table 4.3: Composition of primary motion primitives

Motion primitive	Motion directions	Constraints
Pivot	Object rotation	Ground sticking (both) Hand sticking (both) Hand flush-contact
Slide object left	Object rotation Object translate left	Ground sticking (right) Ground contact Hand sticking (both) Hand flush-contact
Slide object right	Object rotation Object translate right	Ground sticking (left) Ground contact Hand sticking (both) Hand flush-contact
Slide hand left	Object rotation Hand translate left	Hand sticking (right) Hand contact Ground sticking (both) Hand flush-contact
Slide hand right	Object rotation Hand translate right	Hand sticking (left) Hand contact Ground sticking (both) Hand flush-contact
Wall pivoting	Object rotation (wall)	Hand sticking (both) Hand flush-contact
Slide hand left (wall contact)	Hand translate left	Ground sticking w/ wall support Hand flush-contact Environment flush-contact Hand contact Hand sticking (right)
Slide hand right (wall contact)	Hand translate right	Ground sticking w/wall support Hand flush-contact Environment flush-contact Hand contact Hand sticking (left)
Corner-corner pivoting	Object rotation Hand relative rotation	Bounded radial force

4.4 Discussion

Before the estimator has collected any information about the system, the controller operates using a conservative guess for the system's geometry and friction properties. The controller is still able to run reasonably well during this initial phase, as long as the guesses are not completely off. As the estimator builds a more accurate model of the system, the controller is able to execute more aggressive motions.

Because the contact configuration controller uses an impedance target as a proxy for regulating poses and wrenches, it can be thought of as a form of indirect force control (IFC). However, it can also be thought of as a flavor of hybrid force-velocity control (HFVC), because it attempts to regulate the pose along the feasible motion directions, while regulating the wrench along the constraint normals. However, unlike many types of HFVC, our controller does not explicitly partition pose and wrench regulation into orthogonal subspaces (resulting in two independent control problems). Instead, everything is put in the same pot, and we rely on the QP and the low-level impedance controller to solve both problems simultaneously. This design is advantageous, because the compliance introduced by the impedance controller is naturally forgiving of estimation error. This is demonstrated by the ability of the controller to operate using naive guesses of the admissible motion directions. For instance, we usually seed the controller with the initial guess of $\vec{q}_\theta = (0,1)$ for the object pivoting motion.

Why do we use an impedance control layer instead of a stiffness control layer (which emulates a mass-spring system without the damping)? The main reason for choosing impedance control over stiffness control is that the damping provided by impedance control plays an important role in dissipating kinetic energy, which helps stabilize the system, smooth motions, and reduce oscillations. A stiffness controller could work if the lumped system of the robot arm + the stiffness controller is able to quickly dissipate energy. Even then, impedance control is still better, because the damping parameter can be tuned to optimize the behavior of the controller.

On a similar note, it may be possible to implement a version of our contact configuration controller using a position control layer instead of an impedance control layer. For instance, the HFVC framework implemented by Hou et al. for simultaneously regulating motion and contact interactions made use of a position controlled robot [29, 30, 31]. However, there is a significant possibility that using a position controller would result in the coupled system (contact configuration controller + position control layer) becoming unstable. It is difficult to say for sure without trying to implement our system with a position control layer (an experiment which we have not performed). That being said, we usually run our controller using a translational stiffness of 2000 Newtons per meter and a rotational stiffness of 60 Newton-meters, which is compliant in a very tangible way (it is possible to displace the end-effector by up to a centimeter using one finger). I do feel that compliance on this order of magnitude may be essential to allowing the system to be robust to estimation error.

We tuned the stiffness of the impedance controller via trial and error. To do this, we performed a set of experiments in which the end-effector was commanded (while in impedance mode) to follow a circular path at various speeds. From these, we chose a nominal speed that seemed like the maximum desired end-effector speed during manipulation. Given this nominal speed, the stiffness parameters were increased until the system became unstable while attempting to follow the circular path. The stiffness parameters we use in the controller (2000 Newtons per meter for translation and 60 Newton-meters for rotation) are around 1/2 to 1/6th this threshold value. We occasionally test higher or lower values of the stiffness to see how they work with the various iterations of the estimator + controller.

Chapter 5

Experiments and Demonstrations

We perform several experiments and demonstrations to test the performance of our system and show our framework in action. This chapter is split into two parts, each corresponding to the paper that those experiments and demonstrations were included in. In the first paper (Section 5.1), our kinematic estimator was only partially developed, though it was still functional. The second paper (Section 5.2) focused on the improvements to our kinematic estimator and how that expanded the set of motions that the controller could execute.

5.1 Experiments and Demonstrations: Basic Estimator

We conduct several experiments to verify our framework using Franka Emika's Panda robot in impedance control mode. The estimators and controller run at 100 Hz and 30 Hz. The controller parameters, i.e., α, β, and γ, are manually tuned and then fixed for all trials.

Object-agnostic manipulation We test our framework's ability to manipulate a set of unknown polygons and average the results across ten experiments. For each experiment, the robot manipulates either an equilateral triangle, a square, a rectangle, a regular hexagon or a pentagon. The objects' side-length and mass vary between 7 cm to 19 cm and 200 g to 450 g. For two experiments, we add a 500 g mass to the triangle and pentagon. The coefficient of friction between the hand/object and object/ground is measured as approximately 0.5 and 0.8, respectively. The estimator and controller are initialized, and the manipulator moves through a long sequence of contact configurations during each trial. We command

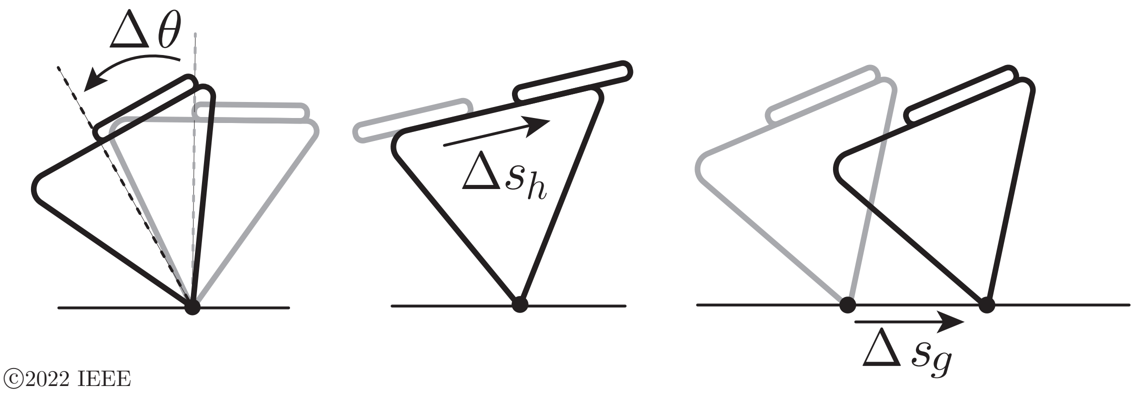


Figure 5-1: Commanded displacements along the admissible motion directions.

changes in θ_o, s_g and s_h in increments of 7.2°, 10 mm, and 10 mm, respectively. A portion of this sequence is shown in Figure 5-2. The most commonly observed failure mode is unintentional slipping at the external contact.

We compare the performance of our kinematic estimator against ground truth provided using the AprilTag vision system [56]. We show an example time trace in Figure 5-3a, and present the estimation error statistics for all ten trials in Figure 5-3b. These results indicate that kinematic estimation is sufficiently accurate for control purposes. We also compare the wrench cone constraint estimates to the contact mode as detected via the ground truth in Figure 5-4. As expected, the measured wrench is near the edge of the friction cone when sliding occurs. We finally analyze the ability of the controller to execute motions along the admissible directions $\Delta\theta_o$, Δs_h, and Δs_g (Figure 5-1). We find that the controller drives the system towards the commanded state (Figure 5-5).

Perturbation rejection We also perform experiments to test the controller's ability to reject perturbations along the three admissible motion directions (Figure 5-6). We find that the system is able to recover from large perturbations in θ_o, s_h, and s_g as long as the initial state can be reached from the perturbed state under the given contact geometry and mode. For instance, for the given system parameters and initial conditions, ground sliding regulation (bottom, Figure 5-6) can only correct perturbations in one direction.

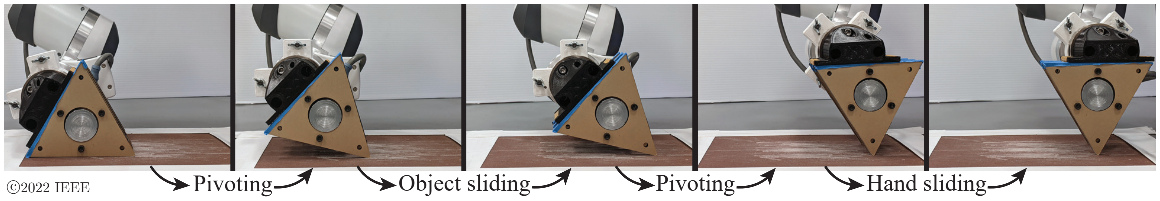


Figure 5-2: A contact configuration sequence during manipulation of an equilateral triangle with an addition of 500 g mass on a high friction surface.

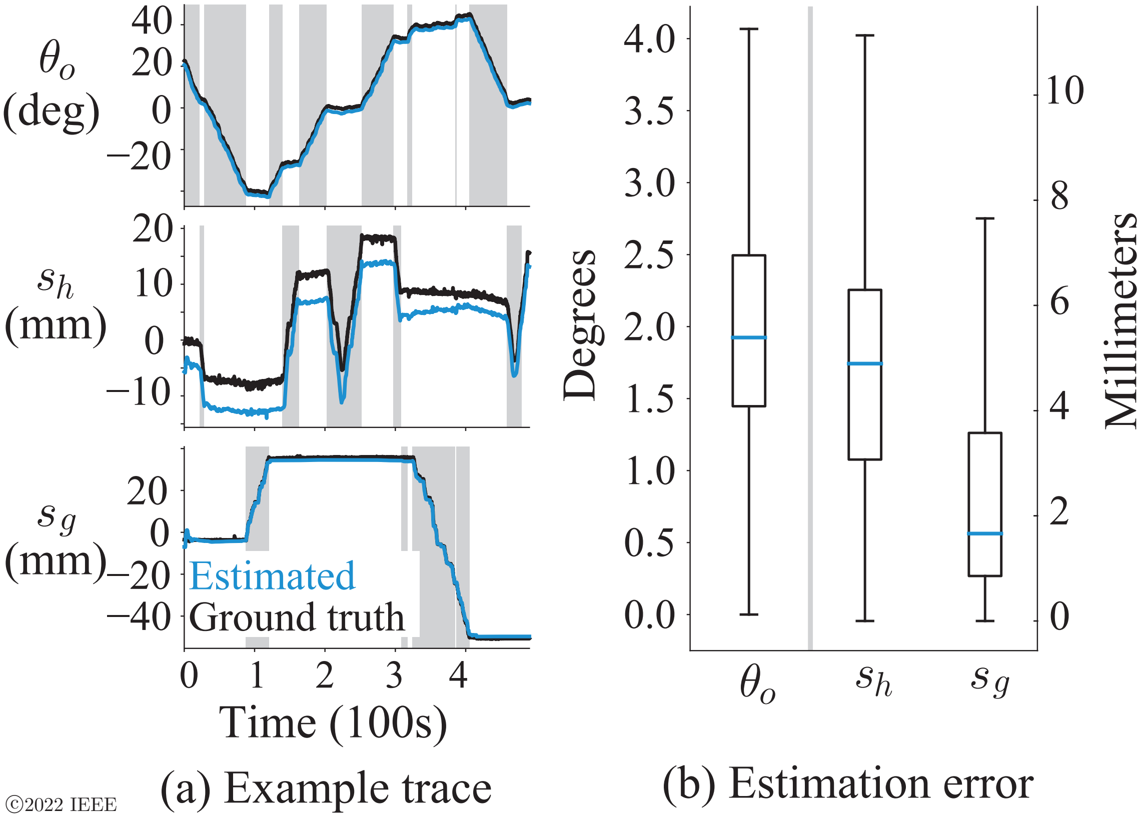


Figure 5-3: (a) Ground truth (black) versus estimated (blue) time traces for object orientation (top), hand sliding position (middle), and object sliding position (bottom) during manipulation of a triangle with no additional mass. Highlighted gray regions indicate when the corresponding variable was commanded to change. (b) Box and whisker plots show absolute estimation error for all ten trials. The median is in blue, top and bottom edges of the box are the 75% and 25% quantiles, and each whisker is 1.5 × the box length. Outliers constitute 5% of the data and are not shown.

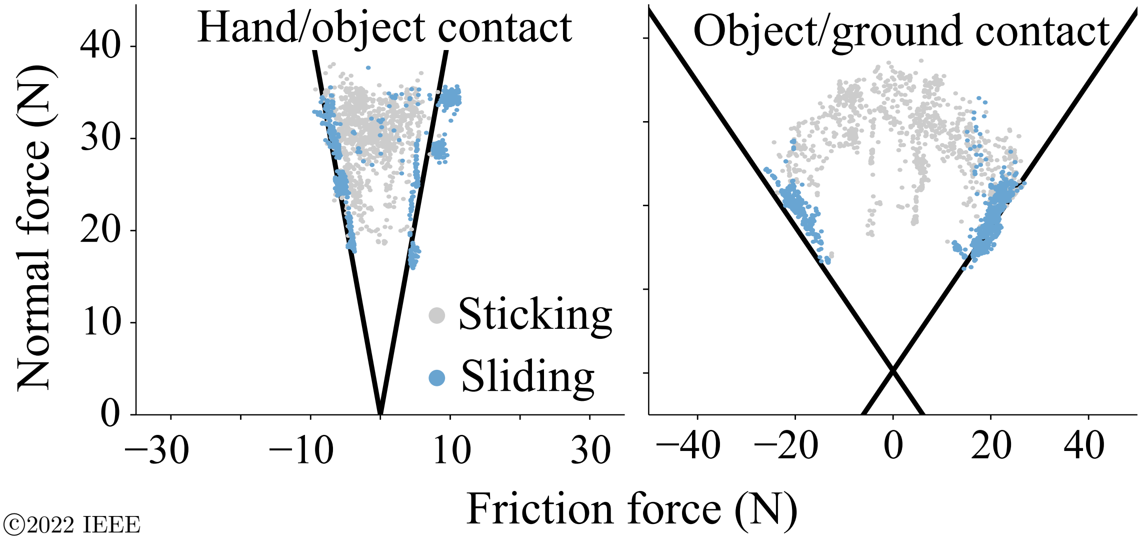
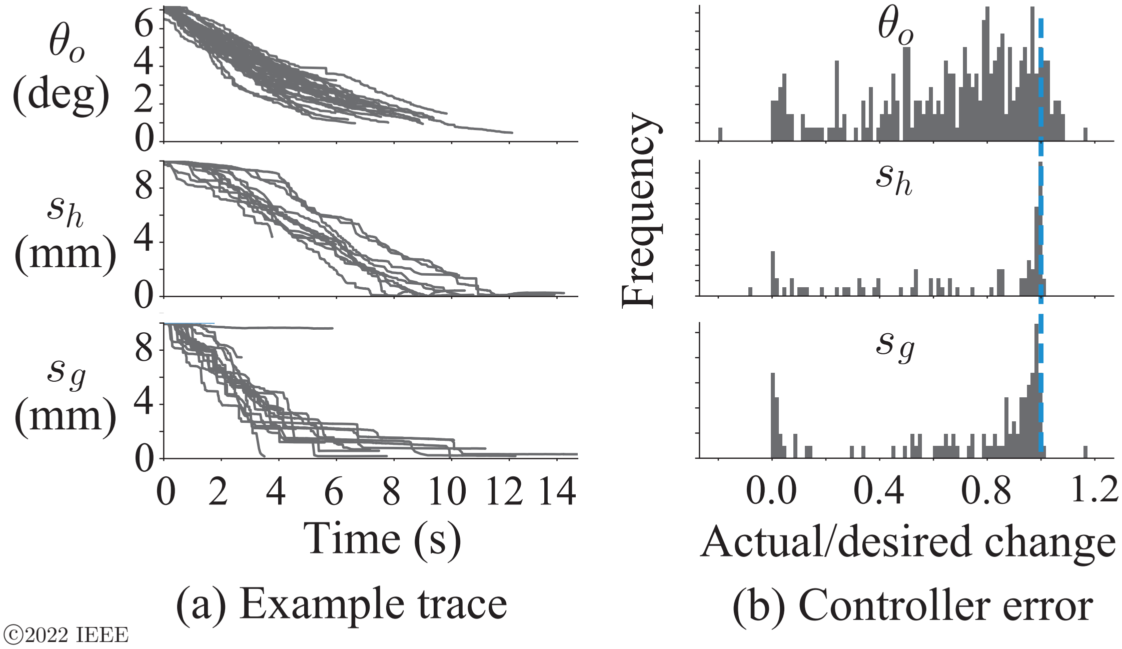


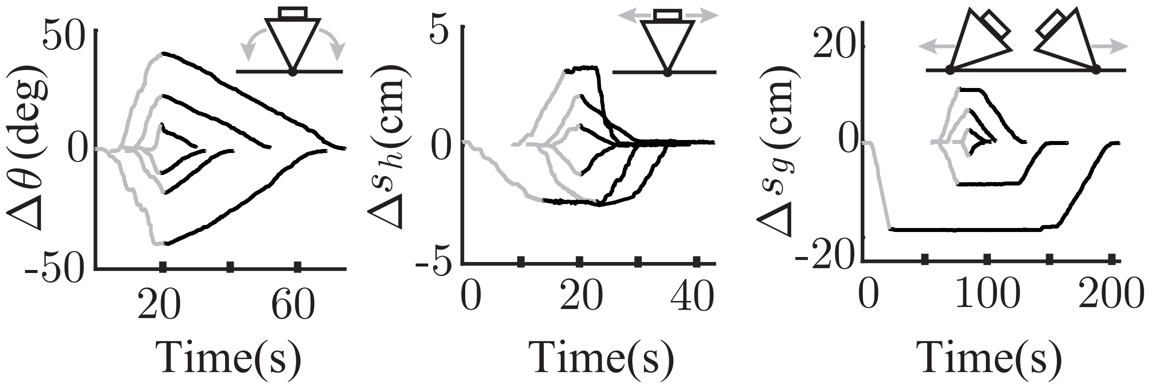
Figure 5-4: Estimated friction cone boundaries (black) for the hand (left) and ground (right) contacts during the same trial as in Figure 5-3. We superimpose measured wrenches, which are colored gray for sticking and blue for sliding based on the contact mode measured via the ground truth.



(a) Example trace

(b) Controller error

Figure 5-5: (a) Time traces of the control error during incremental motions along the admissible motions directions for the same trial as in Figure 5-3. (b) A histogram of the actual change in pose normalized by the commanded change in pose along admissible motion directions across all ten trials. A value of one (dashed blue line) indicates that the commanded change has been fully realized.



(a) Pivoting (b) Hand sliding (c) Object sliding

Figure 5-6: A demonstration of pose regulation along the admissible motion directions. Time traces of the ground truth error in the state variable being actively regulated are shown. Gray lines correspond to a manually applied perturbation, and black lines show the controller's response.

5.2 Experiments and Demos: Advanced Estimator

We perform a set of experiments to validate the completed version of the kinematic estimator. In our setup, we manipulate an object while enforcing contact with a wall (Figure 5-7). Under this contact constraint, the known locations of the ground (y), wall (x), and the object dimensions (w, l) act as ground truth for specific coordinates subsets of the contact locations. The composition of those subsets is determined the object and the type of wall contact. We compare the root mean square error (RMSE) of these coordinate subsets when computed by our vision heuristic (V only), the kinematic estimator making use of live visual feedback (K+V), and the kinematic estimator using a vision prior, but without live visual feedback (K only). The results (Table 5.1) indicate that F/T sensing and robot proprioception are often sufficient for estimating the geometry and locations of the external contacts, which is especially important in situations where the system does not have constant access to visual feedback (object or wall occlusions).

Table 5.1: Kinematic Estimator Tests

Trial	V only	K + V	K only
(a) Wall Pivot	3.28 (cm)	1.68 (cm)	1.66 (cm)
(b) Triangle Pivot	.74 (cm)	.62 (cm)	1.00 (cm)
(c) Object Corner	.38 (cm)	.53 (cm)	.34 (cm)

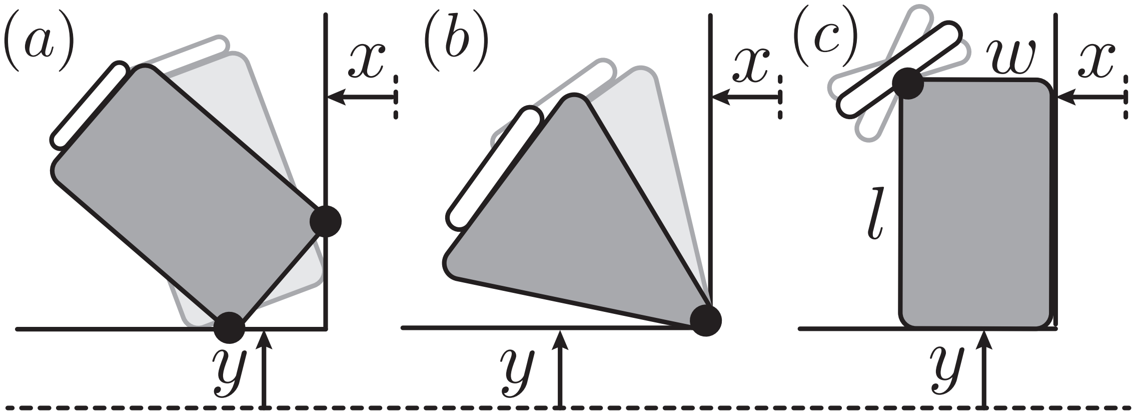


Figure 5-7: Experimental design: we manipulate an object while enforcing wall contact, which provides ground truth for measuring the estimation error. a) Pivoting a rectangle against a corner, and computing the estimation error of the y-coordinate of the bottom vertex and the x-coordinate of the wall contact vertex. b) Pivoting a triangle against a corner, and computing the estimation error of the pivot vertex. c) Pivoting about the vertex of a rectangle that is in flush contact with a wall, and computing the estimation error of the pivot vertex.

To demonstrate the combined capabilities of our estimator and controller, we use them to regulate various shapes through a predetermined sequence of contact geometries (Figure 5-8). For these tests, the kinematic estimator only relies on tactile feedback (after the initial vision prior). We manually jog the hand to disengage from a face/vertex of the object and then reengage at a different face/vertex.

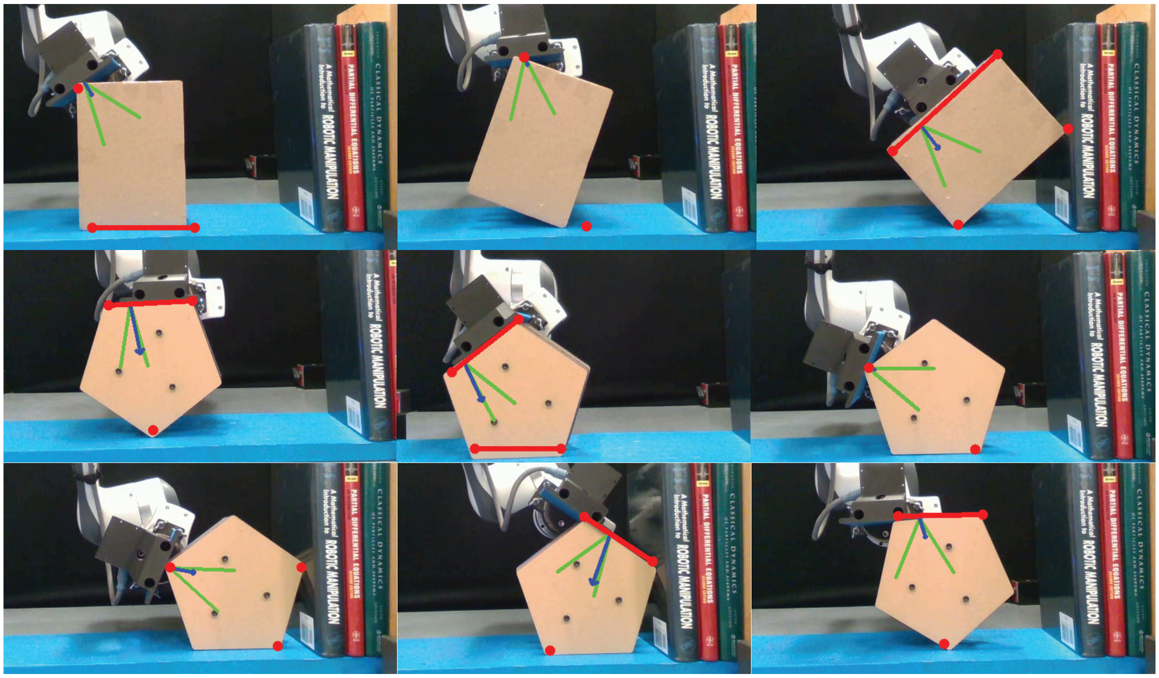


Figure 5-8: Demonstrations: To test our framework, we regulate the system through a sequence of contact configurations. The measured robot wrench (blue) and the robot pose are used to estimate the friction constraints (green) and the contact locations and geometries (red). **Top row:** We execute the corner/corner pivoting primitive to move a rectangle from line contact with the ground to wall contact. **Bottom rows:** We regulate a pentagon through sequence of contact geometry combinations.

Chapter 6

Discussion

6.1 Future Work

There is plenty of room to grow this framework beyond its current state. We made many simplifying assumptions to facilitate modeling, estimation, and control. Relaxing any one of these assumptions allows for a new level of complexity with its own set of unique challenges. I would like to take a moment to explore some of the directions I would take this project if I had more time:

A high level planner: This has been mentioned before, but at the top of my priority list is building a high level planner. This planner would compute the sequence of motion primitives to drive the system from its initial state to some desired pose. Another role of the planner could be to balance the priorities of exploration and exploitation. Upon initialization, the system begins with very conservative guesses of the kinematic and wrench constraints. There are certain exploratory motions that can be executed to quickly expand the wrench constraints and observe the motion directions. It would be great if a planner was designed to focus on exploration early; and then, once it had determined that the frictional and geometric properties had been sufficiently estimated, move on to executing the task at hand. Such a planner would probably make use of the type of motion feasibility analysis shown in Appendix D, computing which primitives that can be successfully executed at each state given the estimated system parameters.

Auto-generation of constraints, motion directions, and primitives: A significant portion of this thesis involves manually deriving the kinematic and motion constraints associated with each contact. On top of this, both the set of costs/constraints in the quadratic program of the controller and the set of variable and constraint factors in the factor graph are hard coded for each motion primitive/discrete contact mode. This amount of hardcoding is acceptable for a system with so few contacts and degrees of freedom. However, as the complexity of the system increases, it is not feasible to enumerate each of the possibilities by hand. Being able to automatically generate (and form compositions of) the kinematic and wrench constraints would allow for the system to be scalable. I believe that this pipeline would consist of two modules. First, a deduction module would identify the active contacts and derive the corresponding kinematic and wrench constraints and admissible motions from a general representation of the object's pose and geometry. Second, a compiler module would construct a motion primitive for each of the admissible motions, and choose the correct subset of wrench constraints to apply in the quadratic program of the contact configuration controller.

The role of learning: Our manipulation framework is entirely model based. We spent more than thirty pages in this thesis describing our model, which then mapped directly to the design of our estimator and controller. There already exists work [42] that is attempting to incorporate machine learning into this process. Given how complex our model was for a simple system, the reality may be that to deal with a system that is significantly more complex, the design of the estimator and controller will have to rely on learning of some kind. Perhaps the ideal case would be a balance, where we can still use a model-based framework to estimate and regulate contact interactions on a small scale, but then use learning to deal with the macro interactions of the system.

Into the third dimension: One of our biggest assumptions is that the system is planar. This greatly simplifies many of the motion and wrench equations. The types of possible contact geometries would expand greatly (instead of point/line vs. flush, we now have patch/patch, patch/edge, patch/point and edge/edge). On top of that, the system has additional degrees of freedom which means more admissible motion direction. In addition to the friction cone, we would need to estimate the limit surfaces associated with any 2D contact patches. It may be possible to reuse most of our framework for simple versions of the 3D problem, for instance pivoting a rectangular prism about its edges (which can just be modeled as a 2D problem anyway). However, to achieve the full range of motions/regulation in 3D, I believe that a significant amount of our framework would need to be rewritten.

Compliance in the object and environment: We rely heavily on the rigid body assumption throughout our framework. The estimator relies on the object maintaining its shape in order to keep track of the location of its boundaries. The controller relies of the kinematic constraints that arise for the rigid body assumption in order to regulate both the object's pose (moving the impedance target parallel to the constraints) and the contact wrench (moving the impedance target normal to the constraints). A limited amount of compliance in the object and environment should not break anything. However, in its current state, you aren't going to get this system to fold clothes or dice vegetables. I think learning task specific policies may be the right approach to highly compliant systems. Trying to use a more sophisticated model of the object and environment that includes stiffness and potential deformations is probably a trap.

Complex object and environment geometries: Similarly, assume that the object is a convex polygon, and the environment consists of static horizontal and vertical lines (for the ground and wall). There are no curved surfaces. There are no concavities. Objects can pivot about their corners, but there is no such thing as rolling. I think, in this case, a better vision system could go a long way. Though our framework primarily only used vision to generate a prior, if vision was used more actively to estimate the shape of the system online, then it may be a lot more feasible to reason about more complex geometries.

Dynamic tasks: Contact configuration regulation has a role to play in the manipulation of dynamic systems. For instance, consider the task of flipping a burger patty. Here, the properties of the spatula/burger contact determine which wrenches and motions can be transmitted to the burger. It would be neat to expand our framework to dynamic tasks like throwing a softball or balancing a mop in the palm of the robot. Unfortunately, I do believe that this would require better hardware. I am not confident with the Panda's ability to execute dynamic tasks accurately (I'm specifically referring to regulating the acceleration of both the object and end-effector). Additionally, we would need to expand our model to reason about inertia and momentum. I think that this could be a tractable problem, provided that more simplifying assumptions were made elsewhere.

Complex robot hands: In our system, the end-effector is a flat palm that is able to move in the vertical plane arbitrarily. Despite its simplicity, the robot is still able to execute a variety of manipulation primitives. Many of the principles in our framework are directly applicable to multi-fingered hands, parallel jaw grippers, and dual armed robots. More contact surfaces means extra actuation. The set of wrench and motions that can be imparted on the object should greatly expand. I'm of the opinion that our contact configuration controller could be directly ported to a dual-arm robot (using a pair of palms as end-effectors), which could greatly expand the set of primitives that could be executed.

Complex system kinematics: One potential research direction could be to regulate systems that consist of longer kinematic chains than a single rigid body. For instance, consider the problem of stacking toy blocks to form a tower. This is a case where the sub-problem of placing a single block looks very similar to the one dealt with in this thesis, but with the added complexity of having to reason about interactions further down the kinematic chain (the contacts interfaces at the base of the tower). A similar problem is packing boxes into a warehouse shelf. Here, each subsequent box would push against the previous. Depending on the state of the system, the previous boxes may slide even further towards the back of the shelf, or they may push up against the back wall, effectively forming a new back wall that must be reasoned about. Once again, the sub-problem of placing a single box looks very similar to our own.

6.2 Lessons Learned

We learned many things during the development of our estimation and control framework. Here, at the end of our journey, I would like to share some of our insights:

Measured wrenches are a reliable predictor of contact behavior: One of our biggest research breakthroughs occurred when we realized that we could accurately predict both point/line vs. flush contact and sticking vs. slipping using the wrench measured by the force-torque sensor. When we compared our observations of the discrete contact state and the measured contact wrench, we noticed that the wrench really was on the interior of the generalized friction cone during sticking-flush contact, and near the boundary during sliding and/or point/line contact.

Before this observation, we spent a significant amount of effort designing a feed forward controller. To this end, we had built a model that would predict how the system would move as a result of a change in the impedance target pose. This approach was doomed to fail for two reasons. First, we were unable to derive any rules that could predict the velocity direction of the system. To be more specific, our model could tell us that both slipping at a contact and object rotation would occur, but it was unable to predict the relative amounts of either motion. More importantly, our model of the underlying impedance controller was only qualitatively correct. Sure, moving the impedance target to the right would result in the end-effector moving to the right (if unconstrained), or the contact force increasing in the $+x$ direction (if constrained), but the increment predicted by the stiffness law did not match with the increment that actually occurred (beyond having the same sign).

Thus, we turned our attention to the measured contact wrench. If it was such a great predictor of contact behavior, then we should make an attempt to close the loop around it. The design of the contact configuration controller was born from this desire: the quadratic program provided a way to regulate the impedance target as a proxy for both the contact wrench and end-effector pose. Once the loop was closed, it did not matter whether or not the impedance controller was able to realize a commanded stiffness: it was sufficient for its qualitative behavior to be correct.

Similarly, our stick/slip estimator was originally built around using robot proprioception as feedback. Once the external contact location had been inferred, we would estimate the slipping velocity at the hand contact, which would then be used to estimate sticking or slipping. Unfortunately, though the pose measurements of the end-effector were accurate, computing the velocity amplified the high frequency noise. Moreover, this method was only feasible if the object never slipped relative to the ground. Once we realized that comparing the measured contact force with the friction cone could accurately predict sticking and slipping at the hand contact, we had the necessary pieces to assemble our stick/slip estimator.

The parallel structure of the estimator and controller: Both the estimator and controller follow the same line of reasoning. We start with a discrete contact state (sticking/slipping and point/line vs. flush contact for each of the contact interfaces). In the case of the estimator, the discrete contact state is estimated through the combination of the friction estimator and the heuristics in the kinematics estimator. In the case of the controller, the discrete contact state is specified by a motion primitive command. The discrete contact state maps to a composition of kinematic and wrench constraints for the system. Both the estimator and controller then convert these constraints into a math problem. In the case of the estimator, this math problem comes in the form of a factor graph, where the constraint factors correspond to the kinematic and wrench constraints, and whose solution maps to the shape and pose of the object. In the case of the controller, this math problem is a quadratic program, where the costs and constraints map to the admissible motions and wrench constraints, and the output maps to the impedance target displacement. The parallel structure of the estimation and control problems is not something unique to our framework. For instance, consider the similarities of the observability and controllability criteria in linear systems, or the similarities between Kalman filters and LQR.

The limitations and benefits of tactile feedback: As discussed in Section 3.4, the prior generated by vision is necessary for the kinematic estimator to function properly. This initial guess does not need to be exact, but it must be reasonably accurate. The fundamental issue here is that it is difficult to disambiguate between possible contact geometries at both the object/hand and object/environment contacts while relying only on tactile feedback. In our implementation, the extra bit of information is provided by the most recent object shape/pose estimate. However, the kinematic estimator has nothing to use when it is first initialized, thus the necessity of the prior. The situation becomes even worse once extra layers of complexity are added to the object and environment (compliance, 3D motion, shapes that aren't polygons, clutter in the environment, etc.). There are limits to how much information that tactile feedback can provide about the system (at least for the form of tactile feedback used in this thesis). Here, vision should prove useful at picking up the slack as the system becomes more and more complex.

That being said, tactile feedback has been excellent at enabling reactive control of the system. Though vision is great at telling us about the global structure of the system, it can be a bit slow. Moreover, it is difficult to infer contact forces from vision alone. On the other hand, robot proprioception and force-torque sensing are lightning fast, which is part of what allows our contact configuration controller to work so well.

Impedance control: The end-effector compliance introduced by the impedance layer is naturally forgiving of estimation error. In fact, the controller is able to reliably function using only a naive guess of the object's shape. This functionality is invaluable during the initial stages of operation, since the controller can enforce stable interactions between the object and robot/environment before the estimator has received enough data to accurately estimate the system parameters.

In addition to this, impedance control provides a natural interface for balancing wrench and pose regulation priorities. Because the contact wrench, end-effector pose, and impedance target pose are linked together, motions of the impedance target can be used to regulate both the contact wrench and end-effector pose simultaneously.

These benefits were discovered in hindsight. Honestly, we stumbled into using an impedance layer as part of our controller: the controller was not designed with these advantages in mind. We had three interfaces available to us: position, force and impedance control. We needed to regulate contact forces, which ruled out position control. On the other hand, it was going to be difficult to command a fixed amount of sliding at a contact using force control. This left us with impedance control, which did the trick.

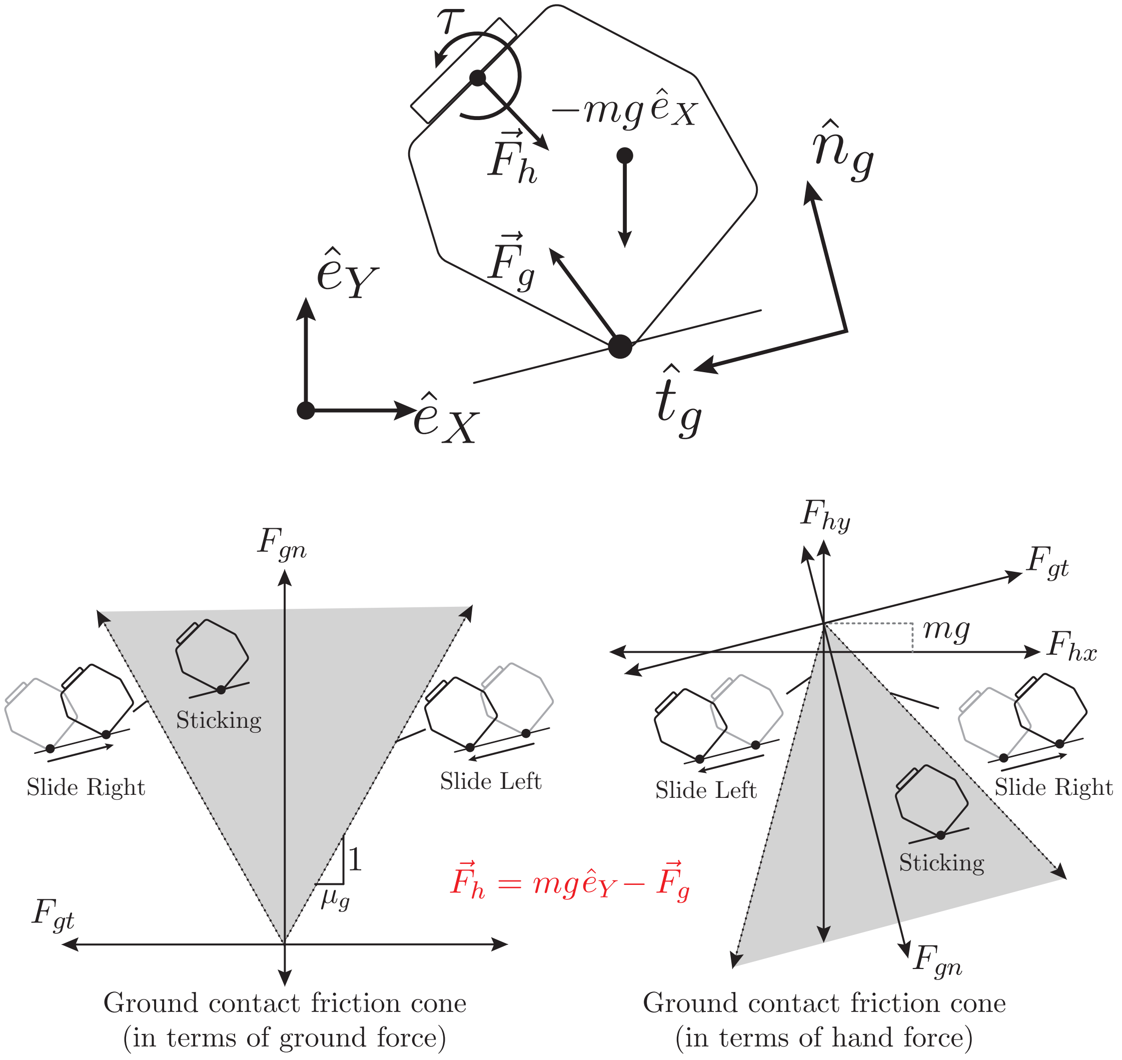
Appendix A

2D Convex Hull Estimator

The original use case of our 2D convex hull estimator was for estimating the friction cone for the ground contact from the measured wrenches extered by the end-effector. At the time, we wanted this estimate to be robust to both the weight of the object, as well as any possible inclination of the ground. The combined effect of the object weight and ground inclination is to translate and rotate the corresponding friction cone in the hand frame (Figure A-1). This proved to be problematic. Unlike the hand contact friction cone (which does not undergo these transformations), trying to directly fit a cone onto the measured contact forces could easily result in an estimate whose feasible region is not entirely contained by the true cone (Figure 3-2). This could lead to the estimator misclassifying sliding contact as sticking, which could compromise the estimator (the reverse misclassification is acceptable however). As such, we needed to approach the problem in a different way.

Our solution was to look at the convex hull of the measured forces. Assuming no measurement error or external perturbations, the measured forces should be entirely contained by the ground contact friction cone, which itself is a convex structure. As such, in the ideal case, the convex hull of the measured forces will be contained by the true ground contact friction cone (Figure 3-2 bottom right).

Computing the convex hull of a set of points is a solved problem in algorithms. There are a billion ways to do it. It's kind of like the sorting problem of computational geometry. Why can't we just copy and paste one of the standard algorithms to compute the exact convex hull of the measured forces, which could then be used as a proxy for the ground



Ground contact friction cone (in terms of ground force)

Ground contact friction cone (in terms of hand force)

Figure A-1: The friction cone for the ground contact expressed in terms of the reaction force of the ground (middle) and the reaction force of the hand (right). The inclination of the ground and the weight of the object rotate and translate the friction cone when expressed in terms of the force of the hand.

contact friction cone? Unfortunately, there are two problems with using the **exact** convex hull of the set of measured forces. First, the exact convex hull is sensitive to outliers. It is entirely possible that, due to either measurement error or an external perturbation of the system, one of the measured contact forces is far outside the true friction cone. Such an outlier would permanently compromise the estimate (Figure A-2 left). We need an estimation method that can reduce the error caused by these outlying points. Second, the purpose of this estimate is to be used as a set of linear constraints in the QP constructed by the contact configuration controller (Section 4.2). Unlike the true friction cone, which is composed of two linear constraints, the convex hull of the measured points could be composed of an arbitrarily large number of linear constraints (Figure A-2 right). This would drastically slow down message passing between the estimator and controller (since there is more information to transmit), and the QP itself. Ideally, our estimation method could extract the underlying polygon (assumed to have a low number of sides) from which our samples are being generated.

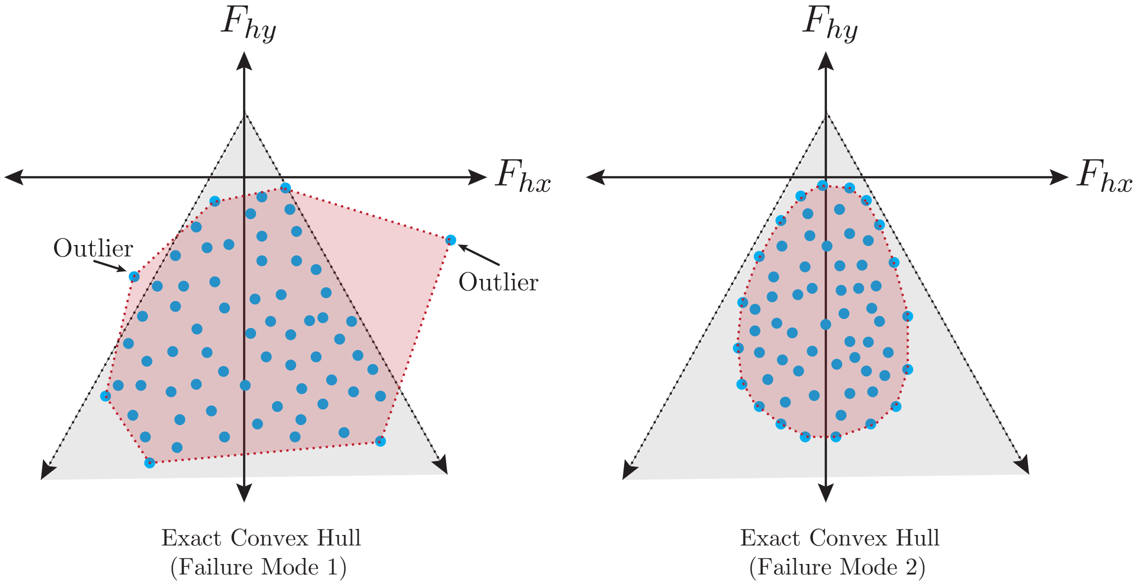


Figure A-2: There are two main issues with using the exact convex hull of the measured forces as a proxy for the friction cone. **Left:** Any outliers (generated by measurement error or external perturbations) can permanently compromise the estimate. **Right:** It is entirely possible for the exact convex hull of the measured point to have an arbitrarily large number of sides, which corresponds to a large number of linear constraints.

In designing our heuristic, we rely on the fact that 1. the underlying structure that we are trying to estimate is a 2D convex polygon with a low number of sides, and 2. each supporting hyperplane of a convex polygon passes through at least one of the polygon's vertices (Figure A-3). Our plan is to estimate the supporting hyperplanes of the measured points for a variety of normal directions. We will then reconstruct the original polygon from these hyperplane estimates.

Algorithm: Let's walk through our algorithm step by step.

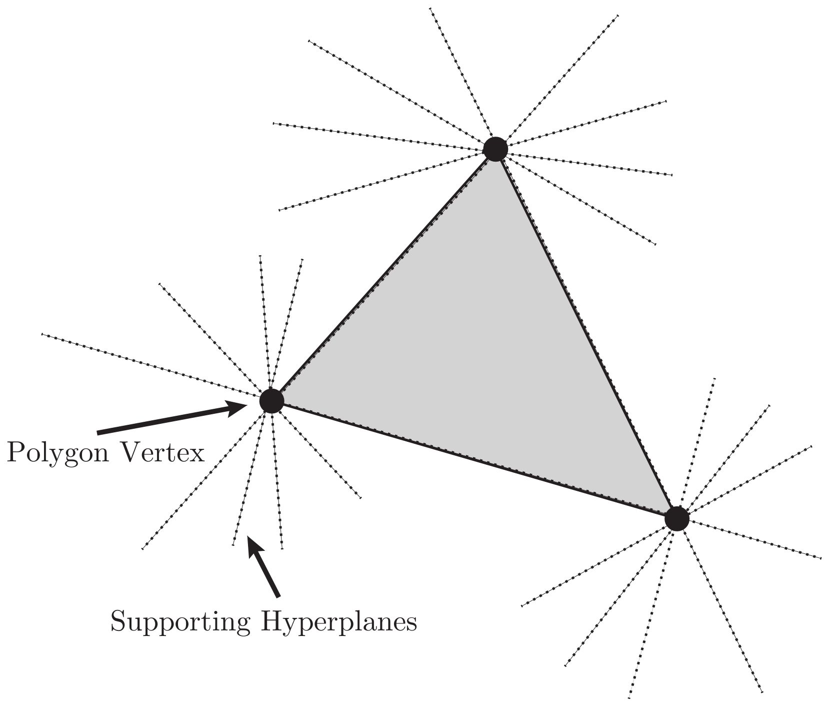


Figure A-3: The supporting hyperplanes of a convex polygon all pass through at least one the polygon's vertices.

Consider a set of measured points:

$$\{\vec{r}_i = (x_i, y_i) | i \in 0, 1, .., n\} \tag{A.1}$$

It should be noted that we will be receiving these points as a continuous data stream. Our algorithm is designed to be online: once a data point has been processed, it can be discarded.

We begin by defining a set of m (fixed) hyperplane normal directions:

$$\left\{ \hat{n}_j = (\cos\theta_j, \sin\theta_j),\ \ \theta_j = \frac{2\pi j}{m} \big| j \in 0, 1, .., (m-1) \right\} \tag{A.2}$$

when the ith measurement is received, we compute the dot product $q_{ij} = \vec{r}_i \cdot \hat{n}_j$ for each of the m hyperplane normal directions. Then, for a given normal direction, we run the result through a slightly modified low-pass filter:

$$b_{ji} = \begin{cases} \min\left(b_{j,i-1} + \Delta b_{max},\ \ \alpha q_{ij} + (1-\alpha) b_{j,i-1}\right) & |q_{ij} \geq b_{j,i-1},\ \ i \geq 0 \\ b_{j,i-1} & |q_{ij} < b_{j,i-1},\ \ i \geq 0 \\ 0 & |i < 0 \end{cases} \tag{A.3}$$

This filter is unidirectional: b_{ji} is monotonically increasing w/respect to i. We achieve this by setting b_{ji} to $b_{j,i-1}$ if $q_{ij} < b_{j,i-1}$, essentially discarding any data points that do not increase the value of b_{ji}. Essentially, b_{ji} approximates the maximum value of $q_{ij} = \vec{r}_i \cdot \hat{n}_j$ w/respect to i, while rejecting any outliers. This is the purpose of the low-pass filter term $\alpha q_{ij} + (1-\alpha) b_{j,i-1}$. In addition, the amount that b_{ji} can increase between measurements is limited to to Δb_{max}. Note that we discard both the measurement, $\vec{r}_i$, and the previous hyperplane offset, $b_{j,i-1}$, once the new hyperplane offset has been computed. At the end of this process, we are left with a set of approximate supporting hyperplanes.

Unfortunately, there is no guarantee that the hyperplane normals will align with the polygon edge normals. As such, if we just take the intersection of all the half-spaces defined by the supporting hyperplanes, the resulting polygon will likely contain several vertices that are outside the boundaries of the original polygon. However, most of the vertices of this “internal" polygon will cluster near the vertices of the original polygon (Figure A-5 bottom left). If the number of supporting hyperplanes is high enough, the number of these external vertices is usually limited to one per face of the original polygon.

What we can do to discard these external vertices is to compute the “star" (I made this term up) of the internal polygon. Specifically, we construct a new shape by connecting vertices that are separated by two edges from one another (Figure A-5 bottom right). If we compute the intersection of the two convex shapes formed by the star, the resulting internal polygon will approximate the original polygon (Figure A-6 top right).

The only issue now is that this second internal polygon can have many sides. We want to select a subset of edges to reconstruct the original polygon. To do this, for each face, we can compute a dimensionless inverse curvature value, γ_k:

$$\gamma_k = \frac{l_k}{P(\theta_{k+1} - \theta_{k-1})} \tag{A.4}$$

where P is the perimeter of the polygon, l_k is the length of the kth edge, and θ_k is the angle of the kth edge normal. The local maximums (w/respect to k) of this inverse curvature function correspond to edges that are close to the faces of the original polygon. These are the subset of edges that we select (Figure A-6 second row), which generates our final estimate of the convex hull (Figure A-6 bottom).

Figures A-7 to A-10 visualize the estimator running on data generated from a set of random shapes. Note how, for the generated data, the exact convex hull falls prey to the two failure modes that we previously mentioned. In the last example, the final estimate was a quadrilateral even though the generating shape was a pentagon (Figure A-10 bottom). This is because two of the sides of the original pentagon were close to parallel, and the estimator accidentally discarded one of them. This face actually did show up as a tiny local maximum in the inverse curvature plot (Figure A-10 top right), however it fell below the threshold parameter and was ignored. This behavior is acceptable given what we need the estimator to do.

It is possible that the original polygon will not entirely contain the final estimate (Figure A-6 bottom). However, we have found that in practice this heuristic has worked pretty well for our purposes. There really isn't any research value here, it's just a neat thing that I made that I wanted to put in my thesis. It does exactly what we need it to do, and that is good enough for me. We also use this convex hull estimator in our vision heuristic to help estimate the vertices of the target object (Appendix B).

As a final note, we have found that using around $32-64$ hyperplanes works pretty well for our friction cone estimator.

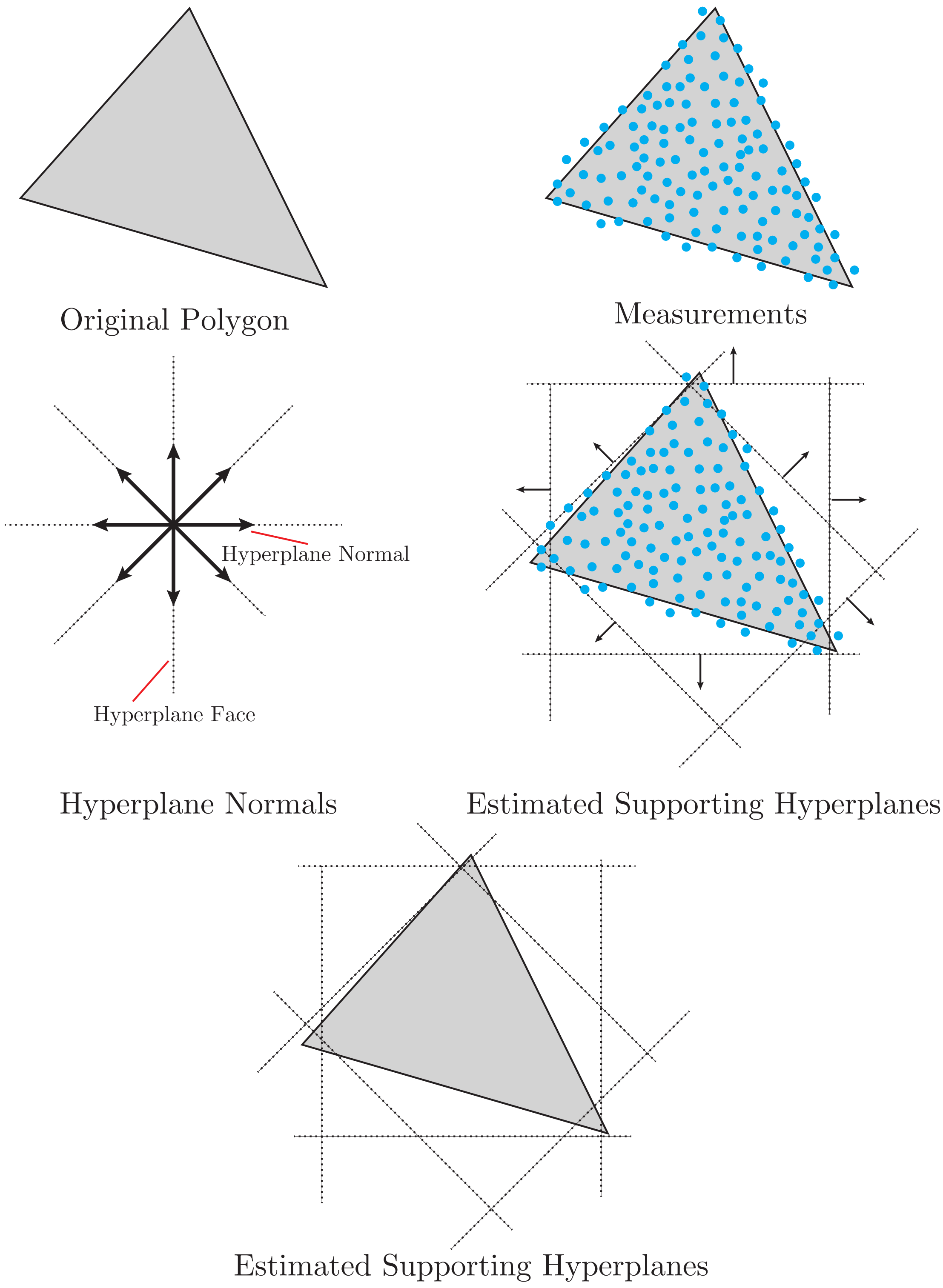


Figure A-4: The first step of estimating the 2D convex hull is to estimate the supporting hyperplanes for a variety of normal directions

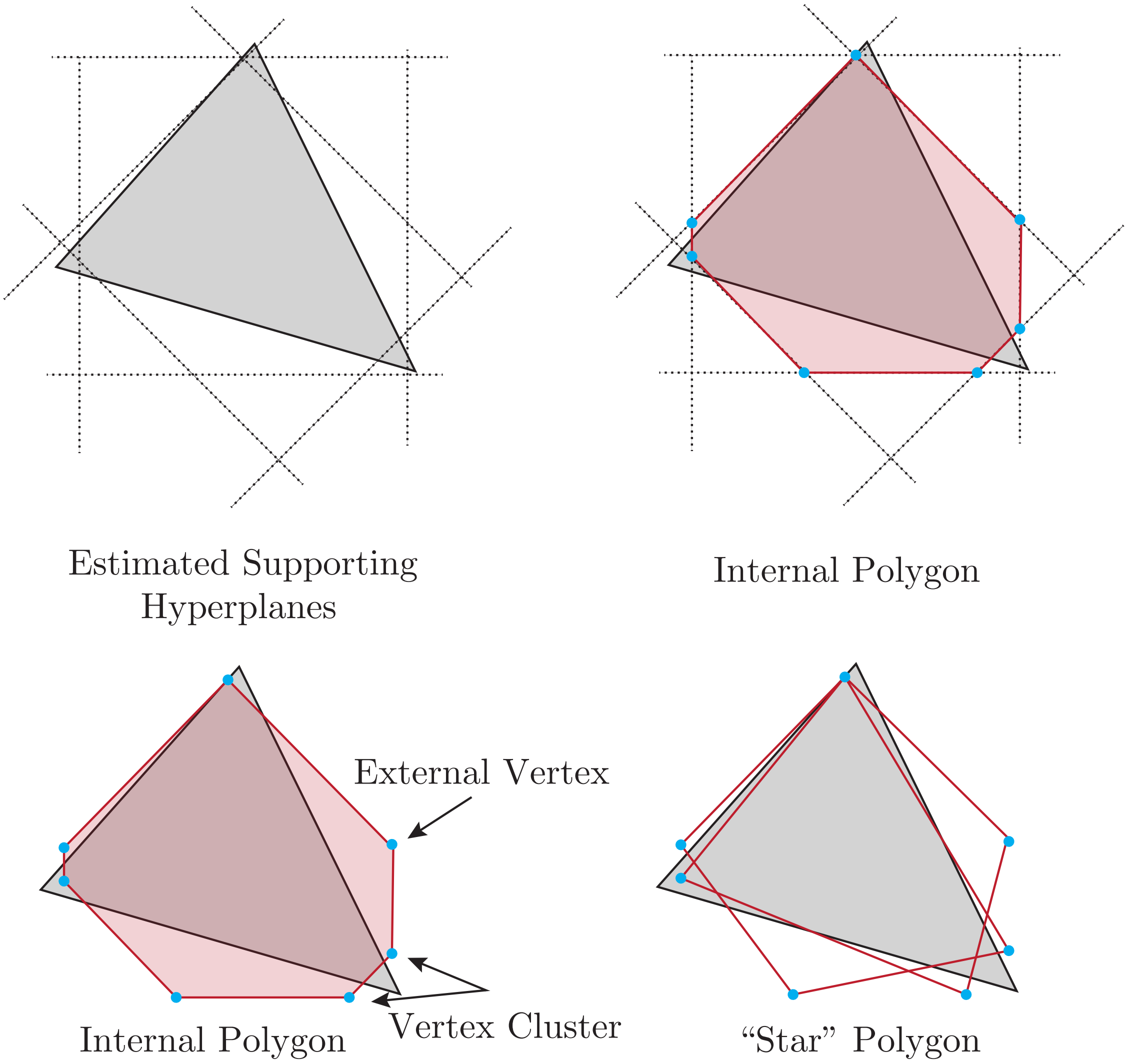
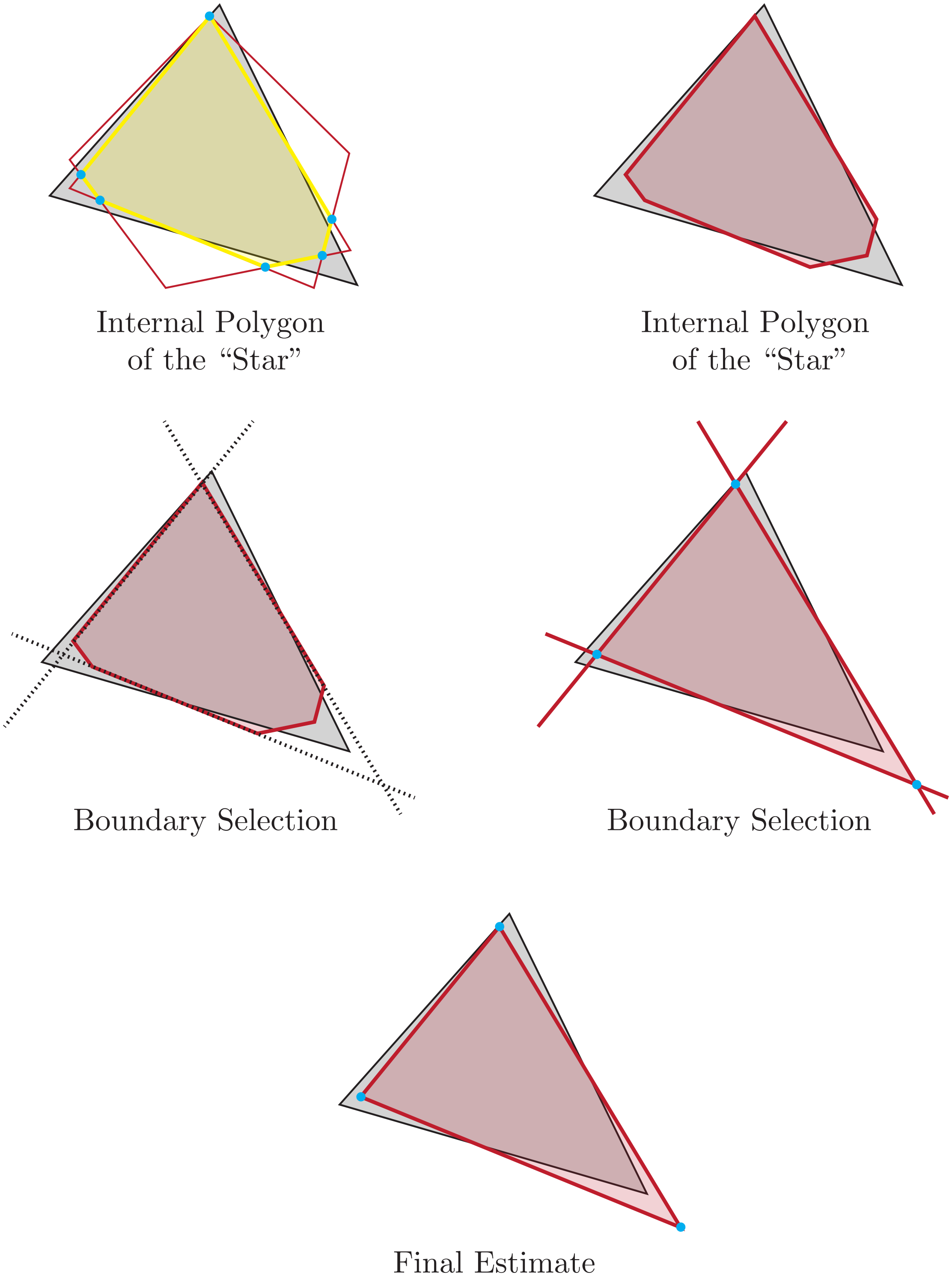


Figure A-5: Once the supporting hyperplanes have been estimated, we compute the polygon describing the intersection of all the half-spaces defined by the hyperplanes. We then compute its “star" (which is just a term I chose, I don’t know of a better thing to call it) by connecting alternating vertices of this internal polygon.



Internal Polygon of the “Star”

Internal Polygon of the “Star”

Boundary Selection

Boundary Selection

Final Estimate

Figure A-6: Once the “star" polygon has been computed, we compute its own internal polygon. From this second internal polygon, we select a subset of boundaries corresponding to the local maximums of the inverse curvature. These are the final boundaries describing our estimate of the original polygon.

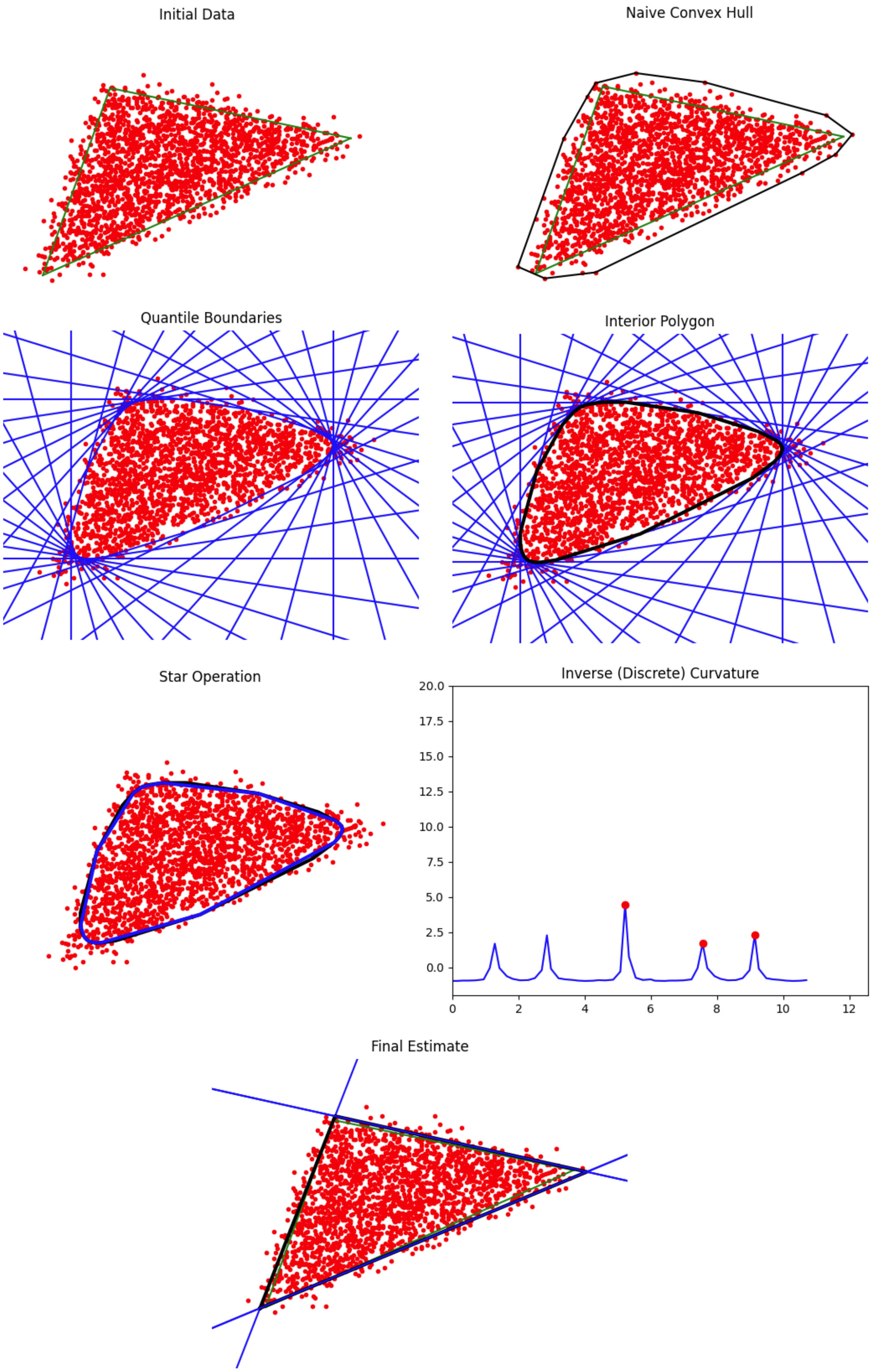


Figure A-7: Estimation of a triangle.

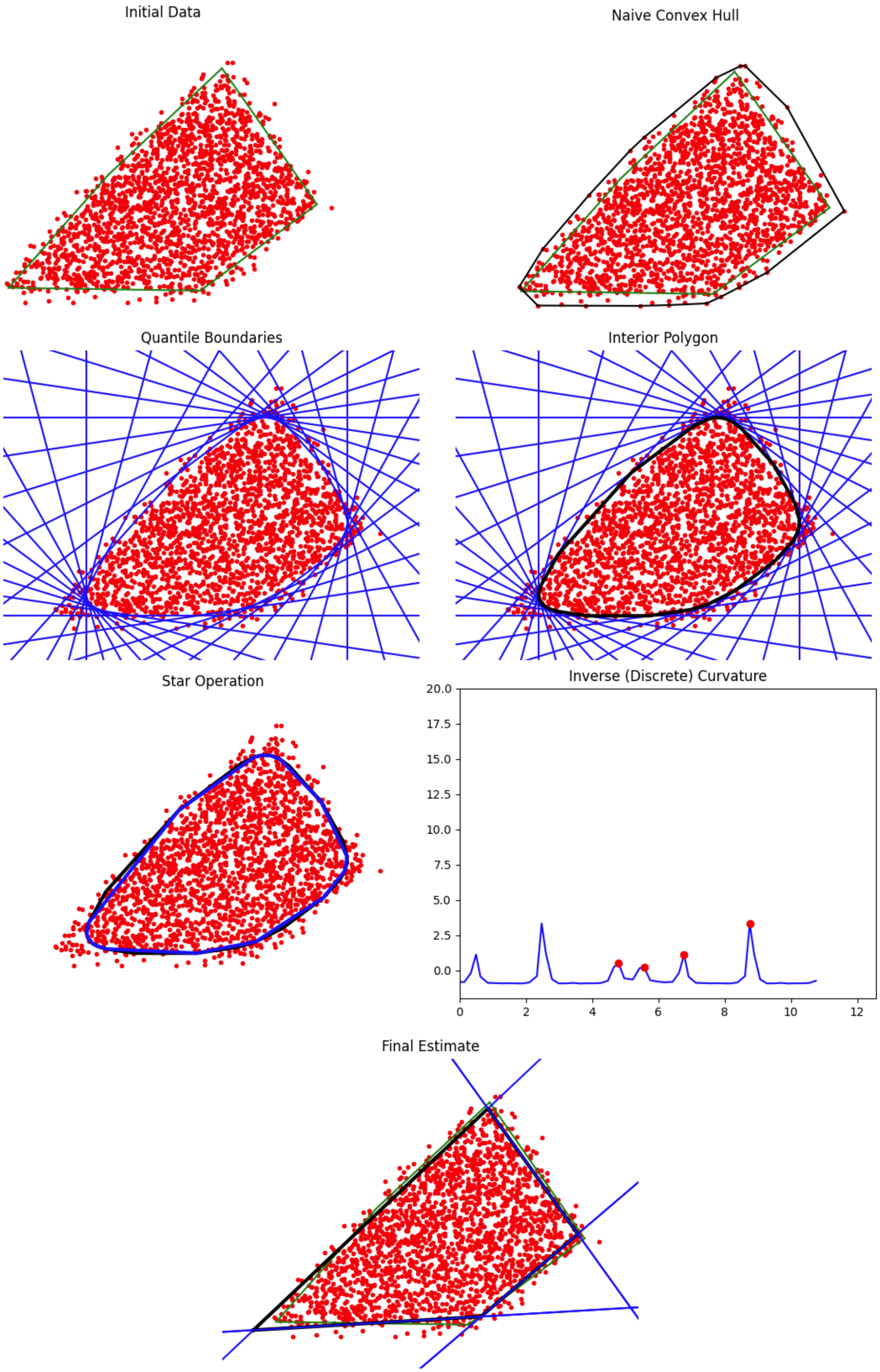


Figure A-8: Estimation of a pentagon.

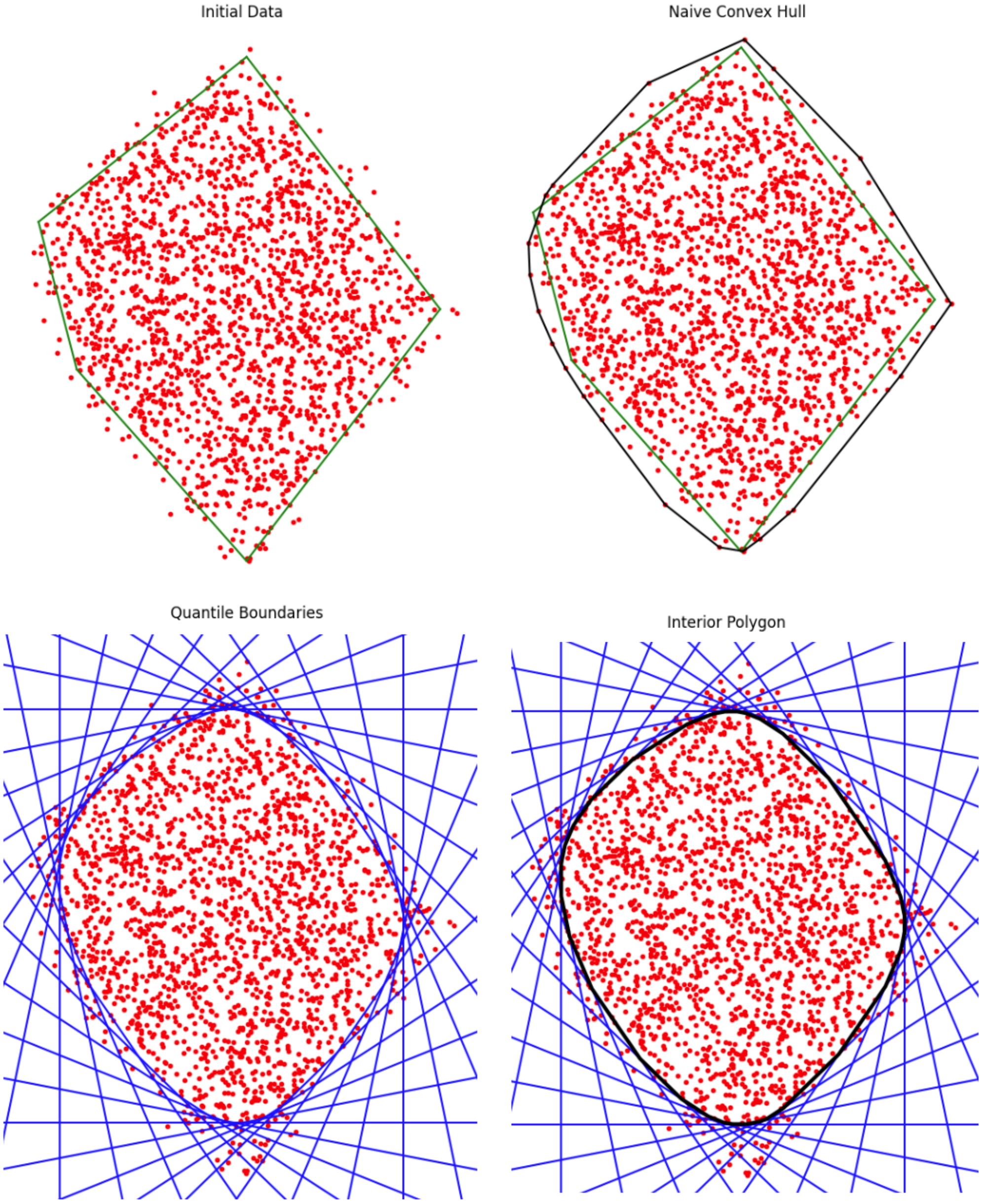


Figure A-9: Estimation of a pentagon.

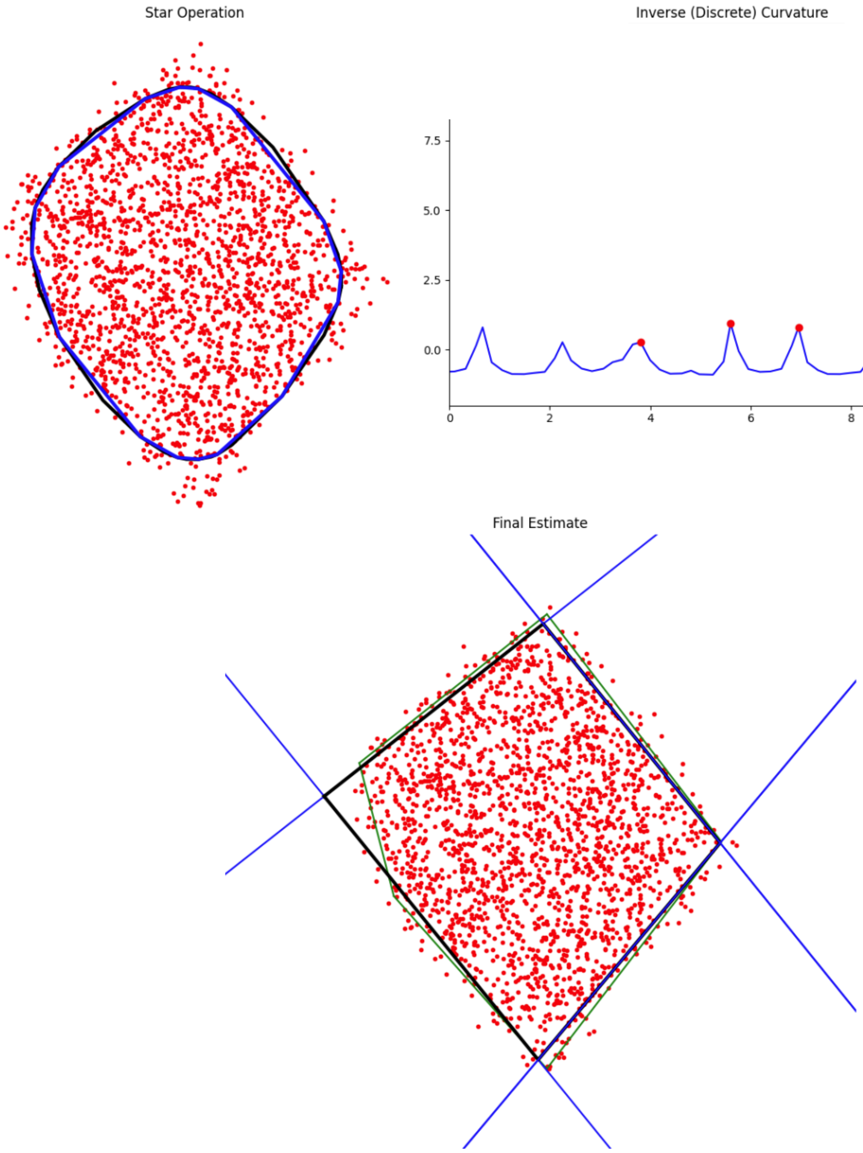


Figure A-10: Continuation of pentagon estimate from previous figure. In this case, the convex hull estimator missed one of the boundaries, with the final estimate being a quadrilateral.

Appendix B

Vision Heuristic

The vision heuristic we use directly estimates the positions of the object vertices in the world-frame, under the assumption that the object is a convex polygon. We start with a few seed pixels that belong to the object's interior (Figure B-2). These seed pixels are centered around either the centroid of the previous estimate, or a fixed distance in the normal direction from the robot palm (if contact is detected).

We assume that the object is a somewhat uniform color that is distinct from its surroundings. As such, we want to identify the connected region of pixels with a similar color as the seed pixels. To accomplish this, starting from the seed pixels, we perform a breadth-first-search on a down-sampled version of the image to sparsely floodfill the portion of the object in the image, giving us a blob that approximately covers the object (Figure B-3). We use a down-sampled image in order to save time (since there are fewer pixels), which is especially useful when processing larger objects.

Going back to the original high-resolution image, we randomly sample pixels near the boundary of this blob (Figure B-4) to check if they should belong to the object's interior (Figure B-5). This gives us a more accurate picture of the object's boundaries. Finally, we extract the object vertices (Figures B-6 and B-7) using our heuristic for approximating the 2D convex hull for a noisy data-set (Appendix A). Figure B-8 shows our vision heuristic running on several test object. This heuristic was sufficiently reliable for developing our kinematic estimator, but should be replaced with a standard perception method during real-world implementation.

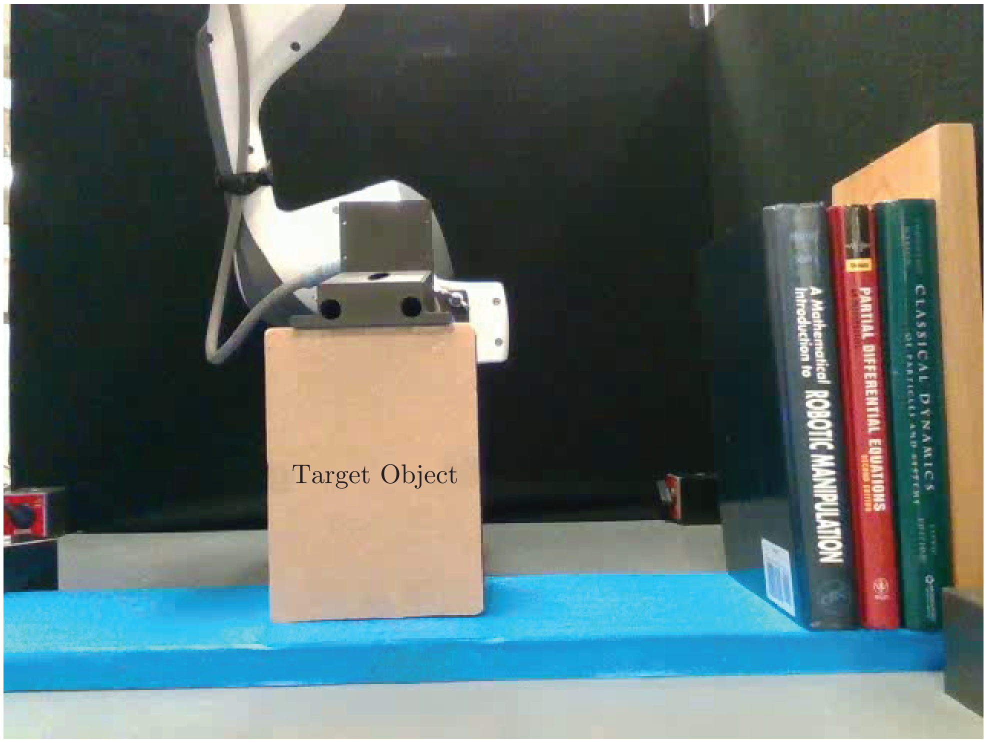


Figure B-1: Starting image. The goal is to extract the object corners from this image.

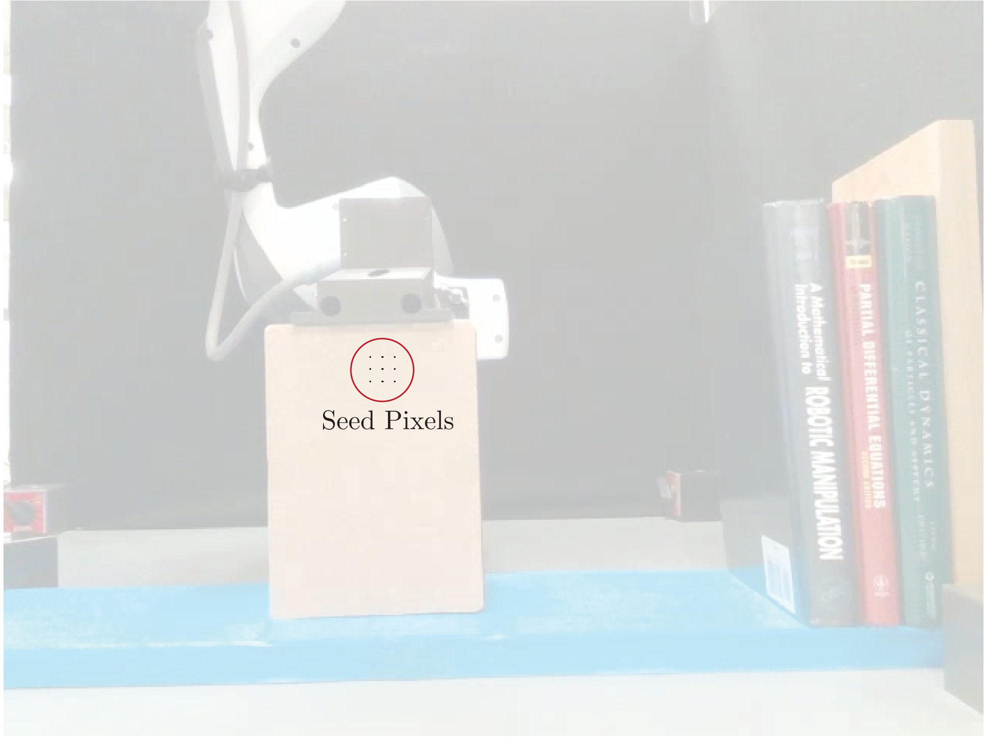


Figure B-2: We begin by identifying a set of seed pixels. These seed pixels are either centered around the centroid of the previous estimate, or some fixed distance in the normal direction of the palm (if contact is detected).

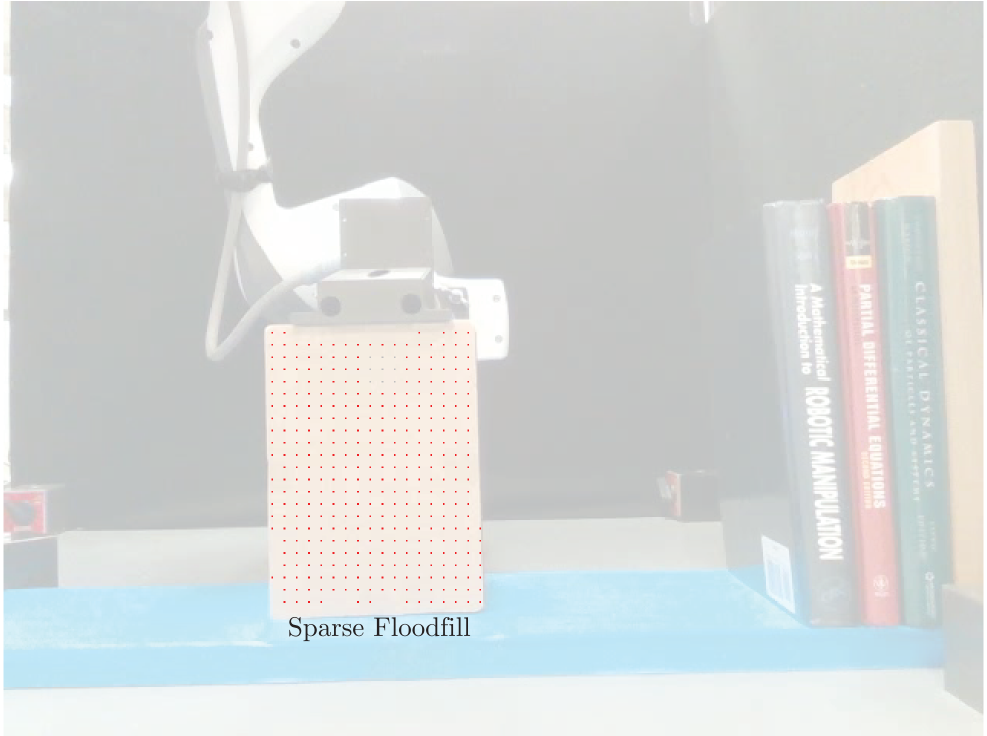

Sparse Floodfill
A Mathematical Introduction to ROBOTIC MANIPULATION
PARTIAL DIFFERENTIAL EQUATIONS
CLASSICAL DYNAMICS

Figure B-3: We perform a breadth-first search to sparsely floodfill the region of pixels with similar values.

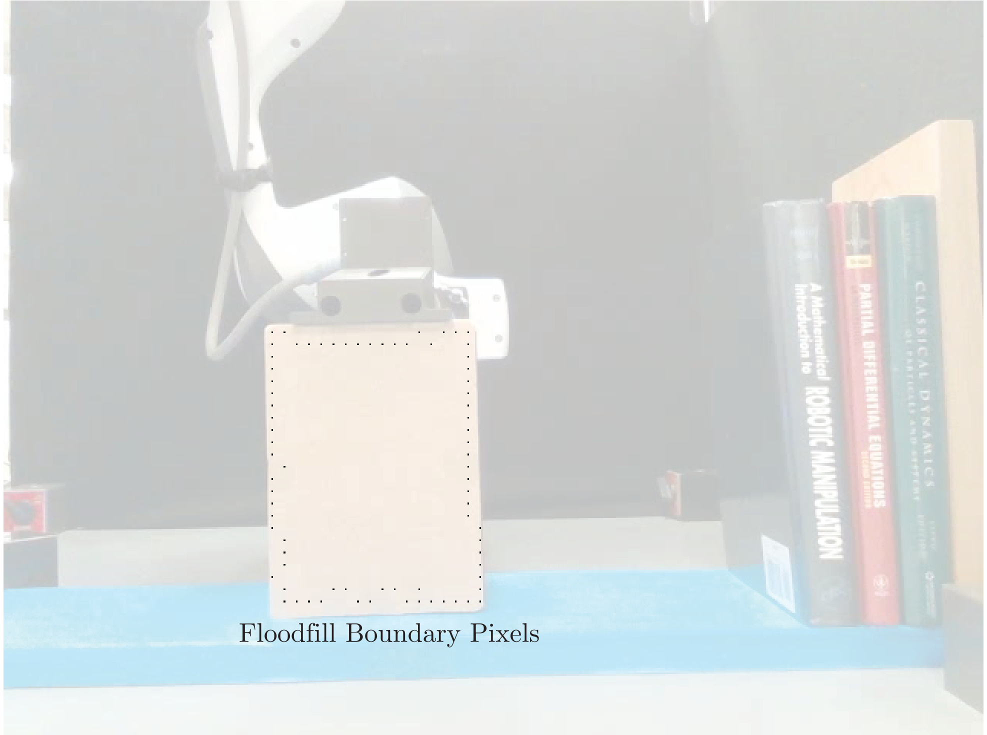

Floodfill Boundary Pixels
A Mathematical Introduction to ROBOTIC MANIPULATION
PARTIAL DIFFERENTIAL EQUATIONS
CLASSICAL DYNAMICS

Figure B-4: We identify pixels on the boundary of the floodfill.

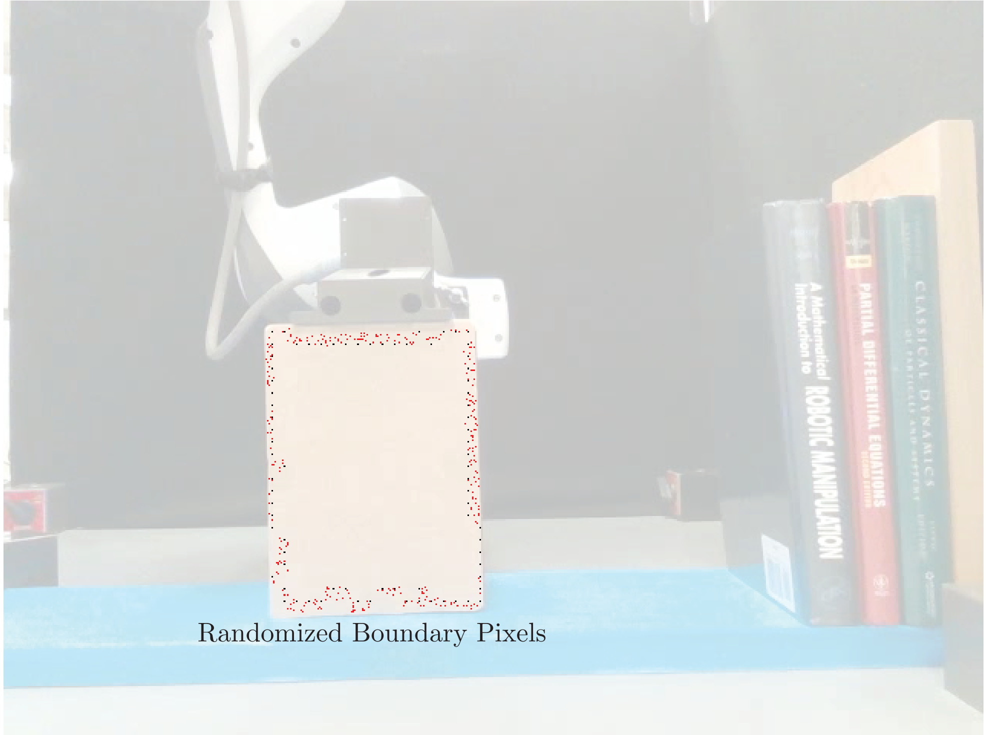

Randomized Boundary Pixels
A Mathematical Introduction to ROBOTIC MANIPULATION
PARTIAL DIFFERENTIAL EQUATIONS
CLASSICAL DYNAMICS

Figure B-5: We randomly test pixels near the boundaries to see if they have the same color.

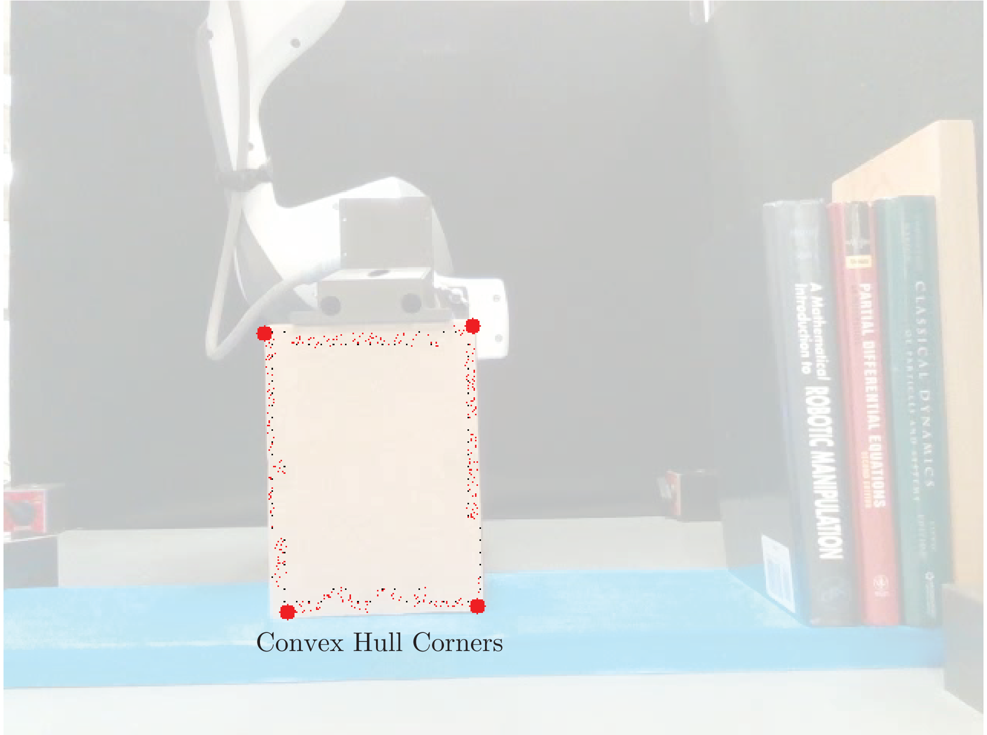

Convex Hull Corners
A Mathematical Introduction to ROBOTIC MANIPULATION
PARTIAL DIFFERENTIAL EQUATIONS
CLASSICAL DYNAMICS

Figure B-6: Finally, we estimate the convex hull of the set of boundary pixels and any randomly tested pixels that were the same color.

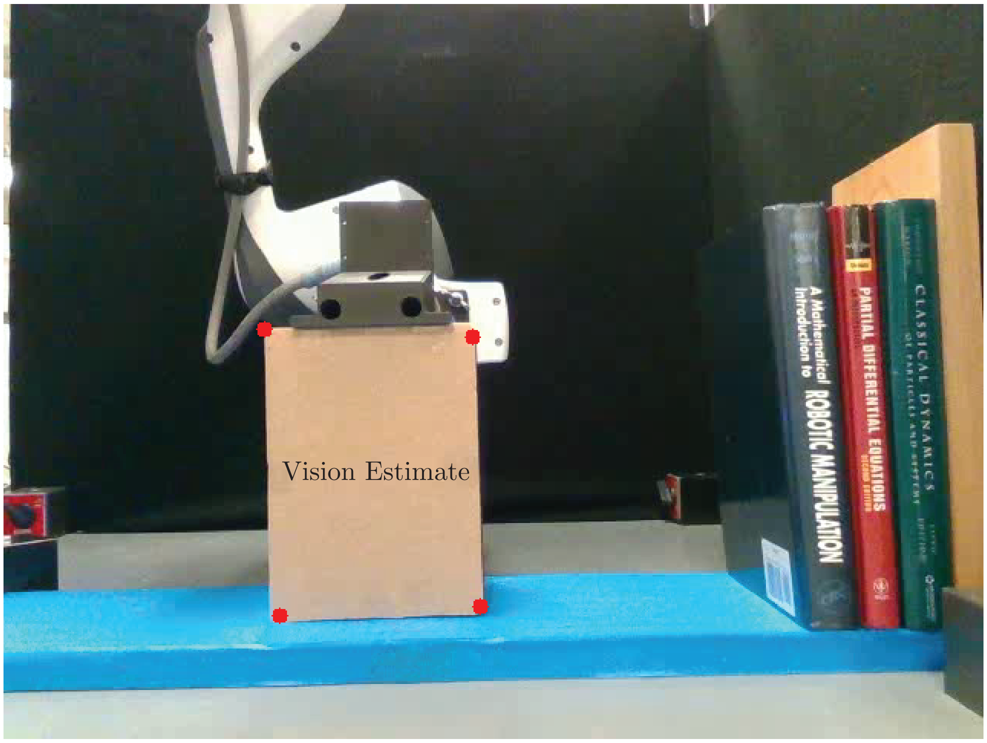

Vision Estimate
A Mathematical Introduction to ROBOTIC MANIPULATION
PARTIAL DIFFERENTIAL EQUATIONS
CLASSICAL DYNAMICS

Figure B-7: The final result is an estimate of the corner locations of the object.

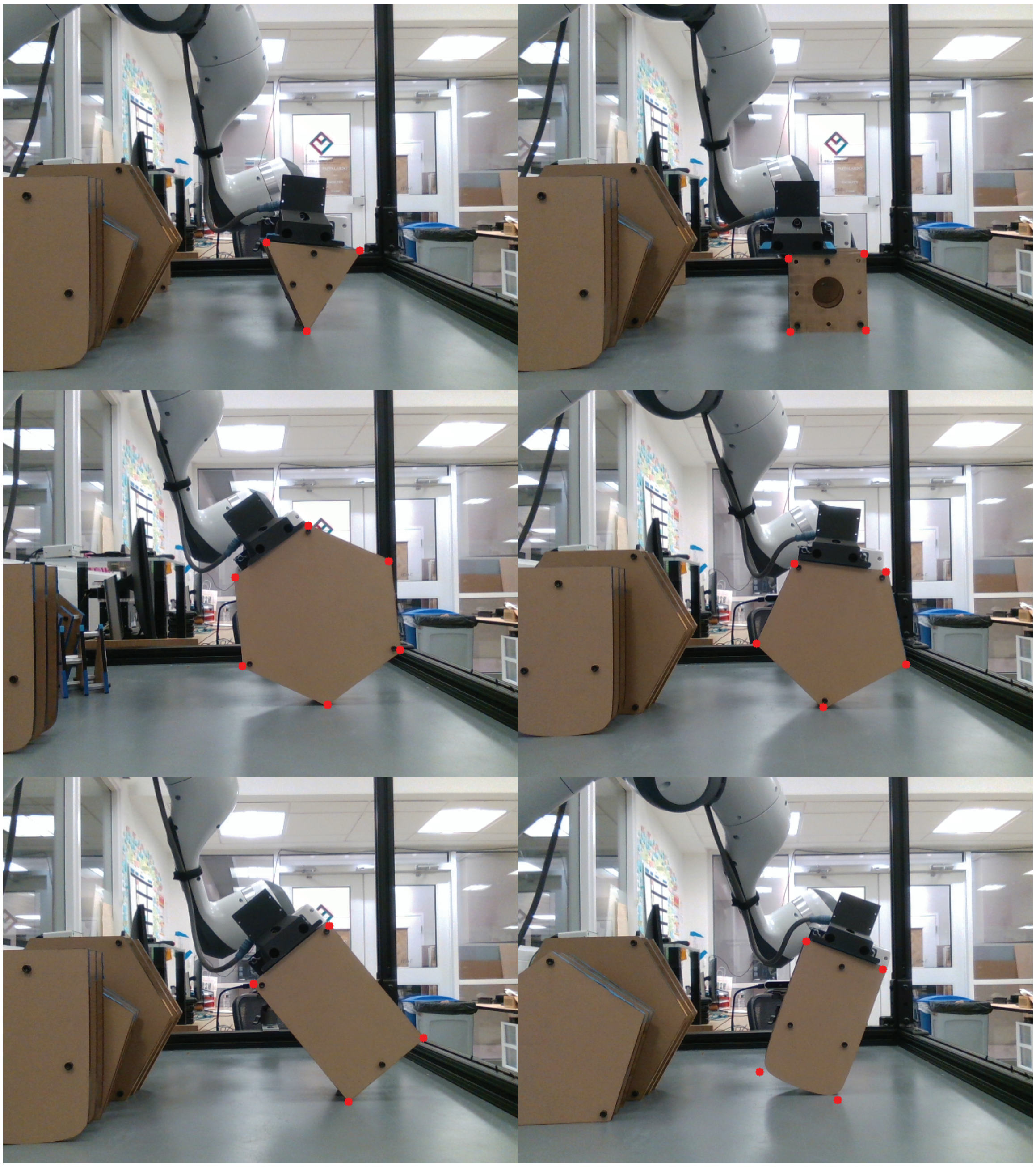


Figure B-8: We demonstrate our vision heuristic on several test objects. Under the right conditions, it can work really well!

Appendix C

Factor Graphs

In this appendix, we will go through a brief overview of factor graphs: what they are and how we use them in the context of this thesis. For a far more thorough treatment on the subject, I refer you to the guide written by Frank Dellaert and Michael Kaess [12].

C.1 What are factor graphs?

Factor graphs are a type of bipartite graph. A bipartite graph has the property where it is possible to color each vertex one of two colors such that no pair of neighboring vertices share the same color. In the case of factor graphs, the two colors correspond to the variables and constraints of a constraint satisfaction problem, with the edges describing the relationships between variables and constraints. As such, factor graphs are a way to visually represent constraint satisfaction problems.

Factor graphs are a popular tool in the SLAM (simultaneous location and mapping) community. In a SLAM problem, a robot builds a map of its environment while simultaneously keeping track of its position in that map. This can be represented as a constraint satisfaction problem, where the variables are the pose of the robot at each time step, and the constraints correspond to odometry measurements (which place a constraint on two consecutive robot poses) or measurements of the position of the robot relative to a landmark in the environment (whose absolute position may or may not be known).

C.2 How are factor graphs used?

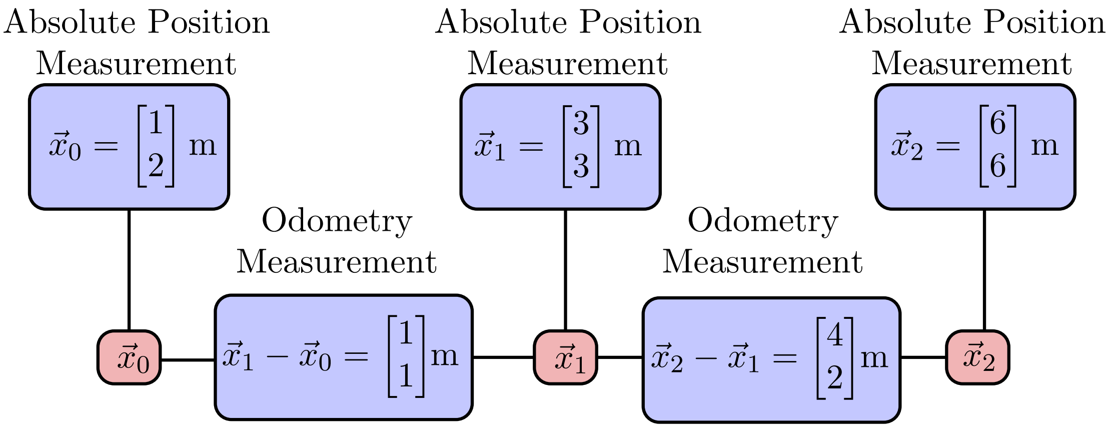

Absolute Position Measurement
$\vec{x}_0 = \begin{bmatrix}1\\2\end{bmatrix}$ m
Absolute Position Measurement
$\vec{x}_1 = \begin{bmatrix}3\\3\end{bmatrix}$ m
Absolute Position Measurement
$\vec{x}_2 = \begin{bmatrix}6\\6\end{bmatrix}$ m
Odometry Measurement
$\vec{x}_1 - \vec{x}_0 = \begin{bmatrix}1\\1\end{bmatrix}$m
Odometry Measurement
$\vec{x}_2 - \vec{x}_1 = \begin{bmatrix}4\\2\end{bmatrix}$m
$\vec{x}_0$
$\vec{x}_1$
$\vec{x}_2$

Figure C-1: An example factor graph depicting the relationships between a robot's position at three instants in time ($\vec{x}_1$,$\vec{x}_2$,$\vec{x}_3$). Here, the variable factors are highlighted in red and the constraint factors are highlighted in blue. Odometry factors constrain the values of two consecutive positions, while absolute position measurements constrain a single variable.

Consider the example illustrated in Figure C-1. Here, $\vec{x}$ describes the position of a robot constrained to move in the plane, with $\vec{x}_i$ being the position of the robot at time step i. In this example, we have three variable factors (shown in red), corresponding to $\vec{x}_0$, $\vec{x}_1$, and $\vec{x}_2$. We also have five constraint factors (shown in blue), corresponding to various odometry and absolute position measurements:

$$\vec{x}_0 = \begin{bmatrix}1\\2\end{bmatrix}\text{ m},\quad \vec{x}_1 = \begin{bmatrix}3\\3\end{bmatrix}\text{ m},\quad \vec{x}_2 = \begin{bmatrix}6\\6\end{bmatrix}\text{ m} \tag{C.1}$$

$$\vec{x}_1 - \vec{x}_0 = \begin{bmatrix}1\\1\end{bmatrix}\text{ m},\quad \vec{x}_2 - \vec{x}_1 = \begin{bmatrix}4\\2\end{bmatrix}\text{ m} \tag{C.2}$$

These constraints can be rewritten as follows:

$$C_0(\vec{x}_0) - \vec{q}_0 = 0, \qquad C_0(\vec{x}_0) = \vec{x}_0, \qquad \vec{q}_0 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \text{m} \tag{C.3}$$

$$C_1(\vec{x}_1) - \vec{q}_1 = 0, \qquad C_1(\vec{x}_1) = \vec{x}_1, \qquad \vec{q}_1 = \begin{bmatrix} 3 \\ 3 \end{bmatrix} \text{m} \tag{C.4}$$

$$C_2(\vec{x}_2) - \vec{q}_2 = 0, \qquad C_2(\vec{x}_2) = \vec{x}_2, \qquad \vec{q}_2 = \begin{bmatrix} 6 \\ 6 \end{bmatrix} \text{m} \tag{C.5}$$

$$C_3(\vec{x}_0, \vec{x}_1) - \vec{q}_3 = 0, \qquad C_3(\vec{x}_0, \vec{x}_1) = \vec{x}_1 - \vec{x}_0, \qquad \vec{q}_3 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \text{m} \tag{C.6}$$

$$C_4(\vec{x}_1, \vec{x}_2) - \vec{q}_4 = 0, \qquad C_4(\vec{x}_1, \vec{x}_2) = \vec{x}_2 - \vec{x}_1, \qquad \vec{q}_4 = \begin{bmatrix} 4 \\ 2 \end{bmatrix} \text{m} \tag{C.7}$$

We can now express this constraint satisfaction problem as a least-squares optimization:

$$\underset{\vec{x}_0, \vec{x}_1, \vec{x}_2}{\operatorname{argmin}} \quad ||C_0(\vec{x}_0) - \vec{q}_0||^2_{W_0} + ||C_1(\vec{x}_1) - \vec{q}_1||^2_{W_1} + ||C_2(\vec{x}_2) - \vec{q}_2||^2_{W_2} \tag{C.8}$$

$$+ ||C_3(\vec{x}_0, \vec{x}_1) - \vec{q}_3||^2_{W_3} + ||C_4(\vec{x}_1, \vec{x}_2) - \vec{q}_4||^2_{W_4} \tag{C.9}$$

Here, the quantity $||\vec{v}||^2_W = \vec{v}^T W^{-1} \vec{v}$ is called the squared Mahalanobis distance. When interpreting a factor graph as a representation of a Bayesian network, W is the covariance matrix describing the noise of a given measurement. However, in the context of our framework, we use W as an optimization weight that can be tuned to increase/decrease the relative importance of a constraint factor in the optimization. Figure C-2 summarizes this process of converting constraints into a set of optimization cost terms.

Factor graphs have a deeper meaning than the interpretation presented thus far. In addition to representing a constraint satisfaction problem, factor graphs simultaneously describe a Bayesian network. The reason each vertex is called a factor is because it represents the probability distribution of a variable (variable factors) or the conditional probability distribution of a measurement given a set variables associated with that measurement (constraint factors). The product of the factors in the graph describe the joint probability of

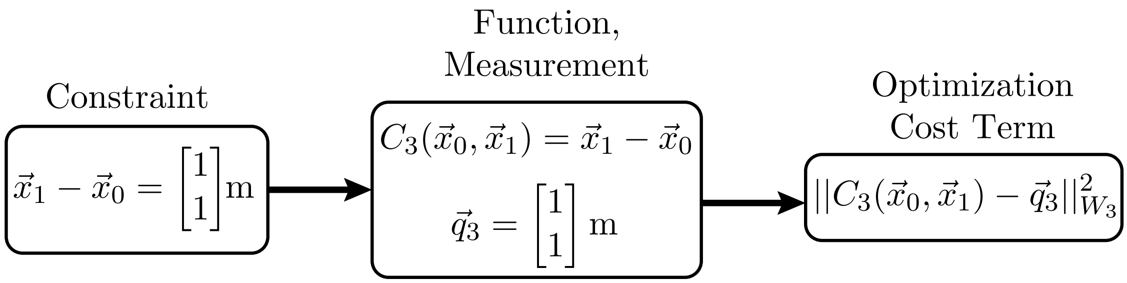

Constraint
$\vec{x}_1 - \vec{x}_0 = \begin{bmatrix}1\\1\end{bmatrix}\text{m}$
Function, Measurement
$C_3(\vec{x}_0, \vec{x}_1) = \vec{x}_1 - \vec{x}_0$
$\vec{q}_3 = \begin{bmatrix}1\\1\end{bmatrix}\text{ m}$
Optimization Cost Term
$||C_3(\vec{x}_0, \vec{x}_1) - \vec{q}_3||^2_{W_3}$

Figure C-2: The process of converting a constraint into an optimization cost term.

the measurements and variables in the network. The least-squares optimization problem corresponds to finding the maximum a posteriori (MAP) estimate of the variables given the measurements. However, for our purposes, we use factor graphs to convert constraint satisfaction problems into least-squares optimization problems. Representing our estimation problem with a factor graph means that we have access to several software toolboxes that can efficiently solve the corresponding constraint satisfaction problem (like GTSAM).

Appendix D

Quasi-static Analysis

In this appendix, we will perform a brief analysis of which actions are feasible for a simplified version of our manipulation system (depicted in Figure D-1).

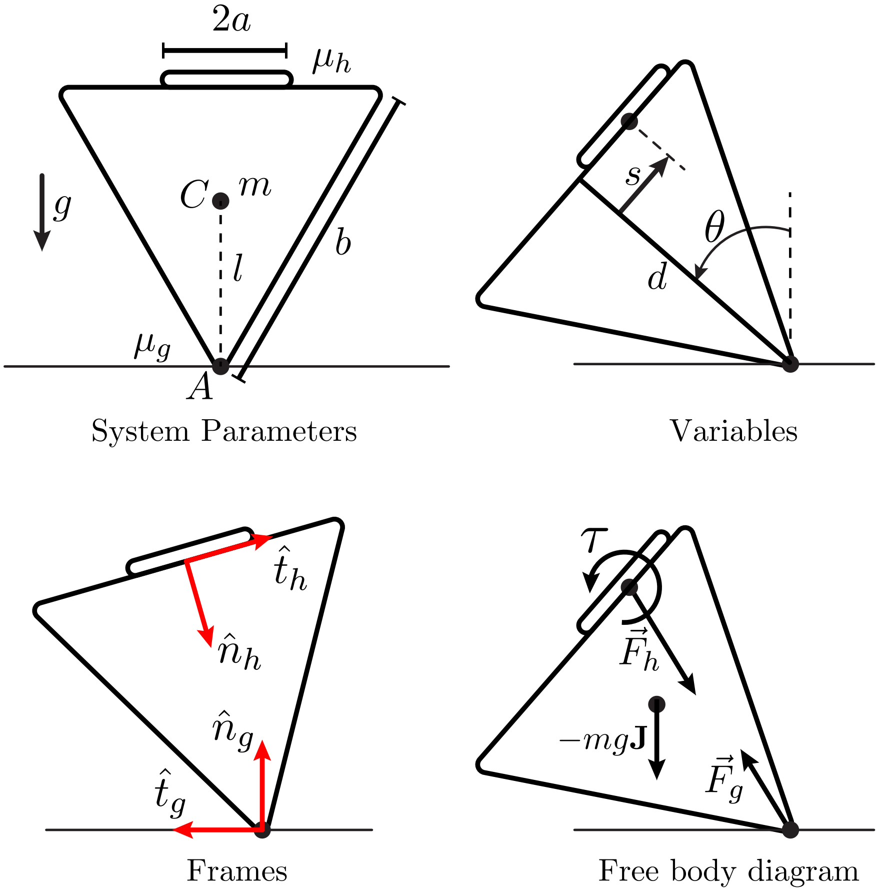


Figure D-1: Simplified system consisting of a palm in contact with an equilateral triangle.

Here, we consider a palm of length $2a$ manipulating an equilateral triangle of side length b, mass m, and height d. The friction coefficients at the hand/object and object/ground contacts are given by μ_h and μ_g respectively. The relative tangential displacement between the hand and the object is given by s. The orientation of both the hand and object (which are assumed to be in flush contact) are given by θ. The unit vectors $(\hat{n}_h, \hat{t}_h)$ and $(\hat{n}_g, \hat{t}_g)$ are normal and tangent to the hand and ground respectively. There are three wrenches acting on the object: the hand wrench $(\vec{F}_h, \tau)$, the ground wrench $(\vec{F}_g, 0)$, and gravity $(-mg\mathbf{J}, \mathbf{0})$.

We begin by deriving the friction constraints for the system. The easiest to analyze is the friction cone for the hand contact, depicted in Figure D-2.

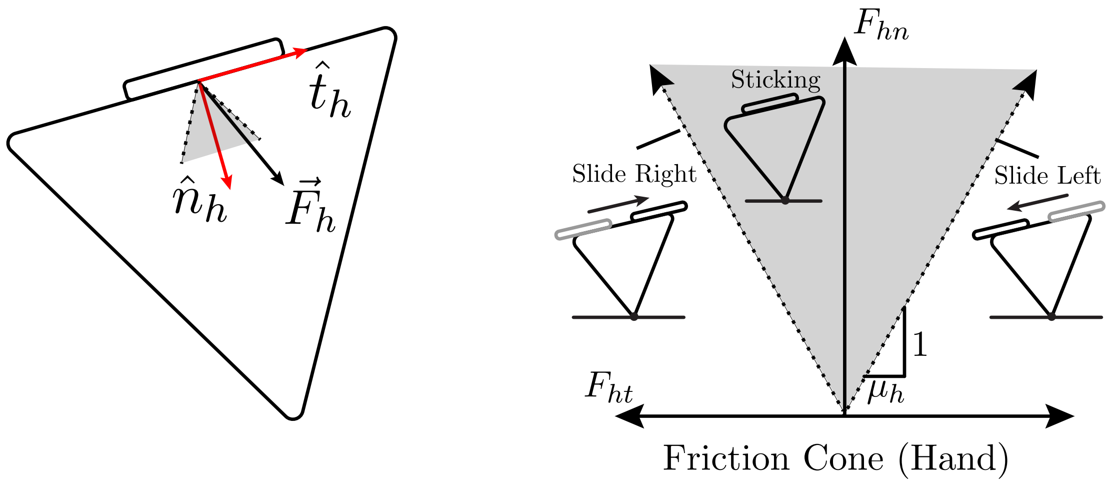


Figure D-2: Friction cone for the hand contact.

Here, we decompose the force exerted by the hand, $\vec{F}_h$, into its normal and tangential components, (F_{hn}, F_{ht}). Applying the Coulomb friction law, we get two inequality constraints:

$$-\mu_h F_{hn} \leq F_{ht} \leq \mu_h F_{hn} \tag{D.1}$$

We can rewrite these two constraints as follows:

$$-\mu_h F_{hn} + F_{ht} \leq 0 \quad \perp \quad \text{(hand slide right)} \tag{D.2}$$

$$-\mu_h F_{hn} - F_{ht} \leq 0 \quad \perp \quad \text{(hand slide left)} \tag{D.3}$$

where $\perp$ denotes the motion that is complementary to the wrench constraint. We now define $w = [F_{hn}, F_{ht}, \tau]^T$ as the wrench exerted by the hand, expressed in the hand contact frame. Substituting w into the previous two inequalities, we get:

$$q_{rh}^T w \leq b_{rh} \quad \perp \quad \text{(hand slide right)}, \qquad q_{rh} = [-\mu_h, 1, 0]^T, \;\; b_{rh} = 0 \tag{D.4}$$

$$q_{lh}^T w \leq b_{lh} \quad \perp \quad \text{(hand slide left)}, \qquad q_{lh} = [-\mu_h, -1, 0]^T, \;\; b_{lh} = 0 \tag{D.5}$$

We now analyze the friction constraints for the ground contact, depicted in Figure D-3.

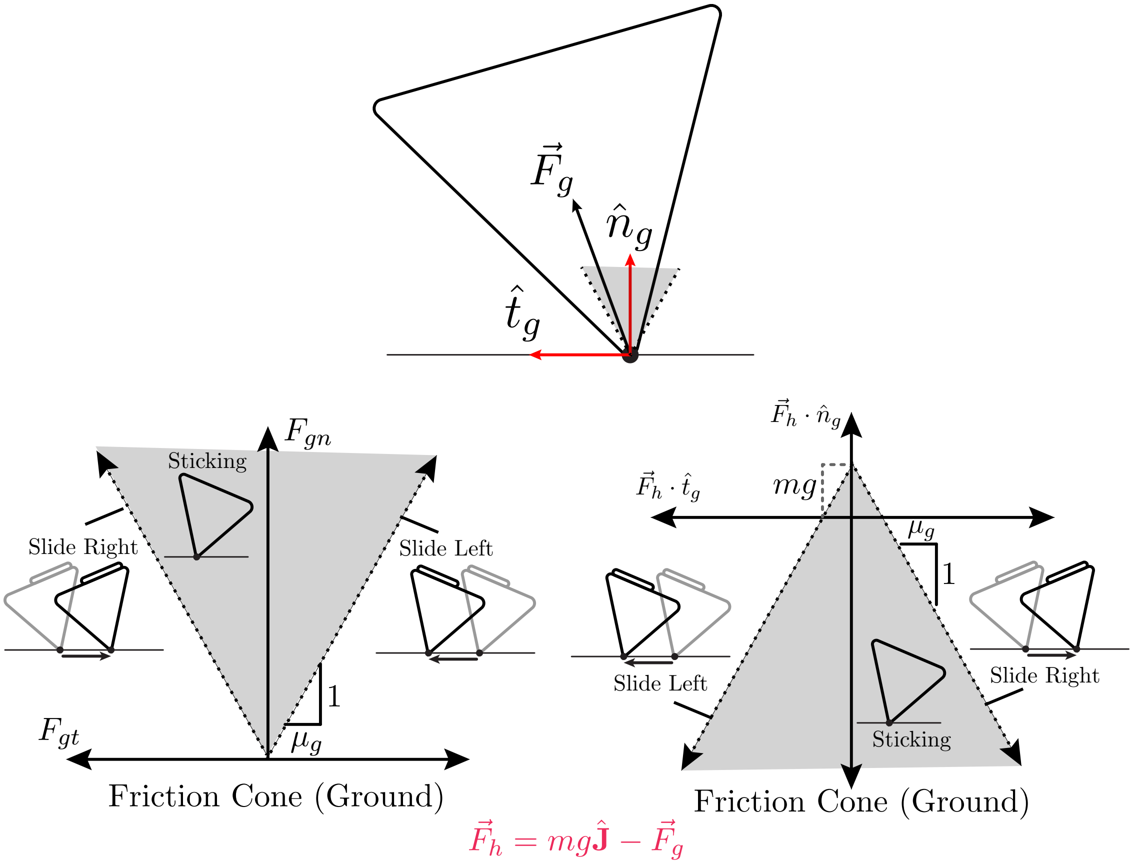


Figure D-3: Friction cone for the ground contact.

We begin by decomposing the force exerted by the ground, $\vec{F}_g$, into its normal and tangential components, (F_{gn}, F_{gt}). Applying the Coulomb friction law, we get two inequality constraints:

$$-\mu_g F_{gn} \leq F_{gt} \leq \mu_g F_{gn} \tag{D.6}$$

We can rewrite these two constraints as follows:

$$-\mu_g F_{gn} + F_{gt} \leq 0 \quad \perp \quad \text{(object slide right)} \tag{D.7}$$

$$-\mu_g F_{gn} - F_{gt} \leq 0 \quad \perp \quad \text{(object slide left)} \tag{D.8}$$

Currently, these inequalities are expressed in terms of the force exerted by the ground. We would like to express them in terms of the force exerted by the hand. To do this, we apply the quasi-static assumption:

$$\vec{F}_h - mg\hat{\mathbf{J}} + \vec{F}_g = 0 \quad \rightarrow \quad \vec{F}_g = -\vec{F}_h + mg\hat{\mathbf{J}} \tag{D.9}$$

$$F_{gn} = -\vec{F}_h \cdot \hat{n}_g + mg, \qquad F_{gt} = -F_h \cdot \hat{t}_g \tag{D.10}$$

Substituting this into our friction constraints, we get:

$$-\mu_g(-\vec{F}_h \cdot \hat{n}_g + mg) + (-F_h \cdot \hat{t}_g) \leq 0 \quad \perp \quad \text{(object slide right)} \tag{D.11}$$

$$-\mu_g(-\vec{F}_h \cdot \hat{n}_g + mg) - (-F_h \cdot \hat{t}_g) \leq 0 \quad \perp \quad \text{(object slide left)} \tag{D.12}$$

$$\mu_g(\vec{F}_h \cdot \hat{n}_g) - (F_h \cdot \hat{t}_g) \leq \mu_g mg \quad \perp \quad \text{(object slide right)} \tag{D.13}$$

$$\mu_g(\vec{F}_h \cdot \hat{n}_g) + (F_h \cdot \hat{t}_g) \leq \mu_g mg \quad \perp \quad \text{(object slide left)} \tag{D.14}$$

Ideally, we would like to express these inequalities in terms of the normal and tangential components of the force exerted by the hand, F_{hn}, F_{ht} (as depicted in Figure D-4). To do this, we make use of the following dot products:

$$\hat{t}_h \cdot \hat{t}_g = -\cos\theta, \quad \hat{t}_h \cdot \hat{n}_g = \sin\theta, \quad \hat{n}_h \cdot \hat{t}_g = -\sin\theta, \quad \hat{n}_h \cdot \hat{n}_g = -\cos\theta \tag{D.15}$$

Plugging $\vec{F}_h = F_{hn}\hat{n}_h + F_{ht}\hat{t}_h$ into our friction constraints, we get:

$$\mu_g(F_{hn}\hat{n}_h + F_{ht}\hat{t}_h)\cdot\hat{n}_g - (F_{hn}\hat{n}_h + F_{ht}\hat{t}_h)\cdot\hat{t}_g \leq \mu_g mg \quad \perp \quad \text{(object slide right)} \tag{D.16}$$

$$\mu_g(F_{hn}\hat{n}_h + F_{ht}\hat{t}_h)\cdot\hat{n}_g + (F_{hn}\hat{n}_h + F_{ht}\hat{t}_h)\cdot\hat{t}_g \leq \mu_g mg \quad \perp \quad \text{(object slide left)} \tag{D.17}$$

$$(-\mu_g\cos\theta + \sin\theta)F_{hn} + (\mu_g\sin\theta + \cos\theta)F_{ht} \leq \mu_g mg \quad \perp \quad \text{(object slide right)} \tag{D.18}$$

$$(-\mu_g\cos\theta - \sin\theta)F_{hn} + (\mu_g\sin\theta - \cos\theta)F_{ht} \leq \mu_g mg \quad \perp \quad \text{(object slide left)} \tag{D.19}$$

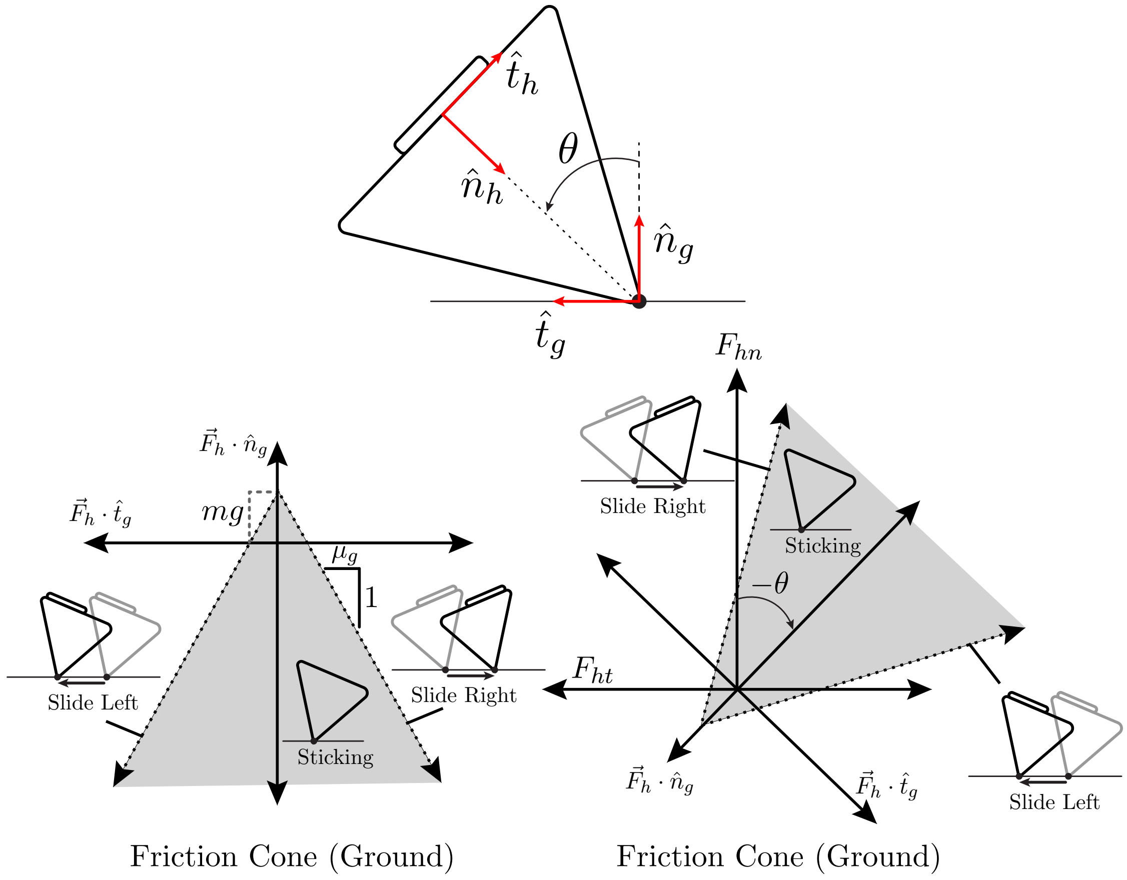


Figure D-4: Friction cone for the ground contact.

We can now express these inequalities in terms of the hand contact wrench, $w = [F_{hn}, F_{ht}, \tau]^T$:

$$q_{rg}^T w \leq b_{rg} \quad \perp \quad \text{(object slide right)}, \qquad q_{rg} = R^T(\theta) p_{rg}, \ \ b_{rg} = \mu_g mg \tag{D.20}$$

$$q_{lg}^T w \leq b_{lg} \quad \perp \quad \text{(object slide left)}, \qquad q_{lg} = R^T(\theta) p_{lg}, \ \ b_{lg} = \mu_g mg \tag{D.21}$$

$$p_{rg} = \begin{bmatrix} -\mu_g \\ 1 \\ 0 \end{bmatrix}, \quad p_{lg} = \begin{bmatrix} -\mu_g \\ -1 \\ 0 \end{bmatrix}, \quad R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{D.22}$$

These are all of the friction constraints that we need to consider.

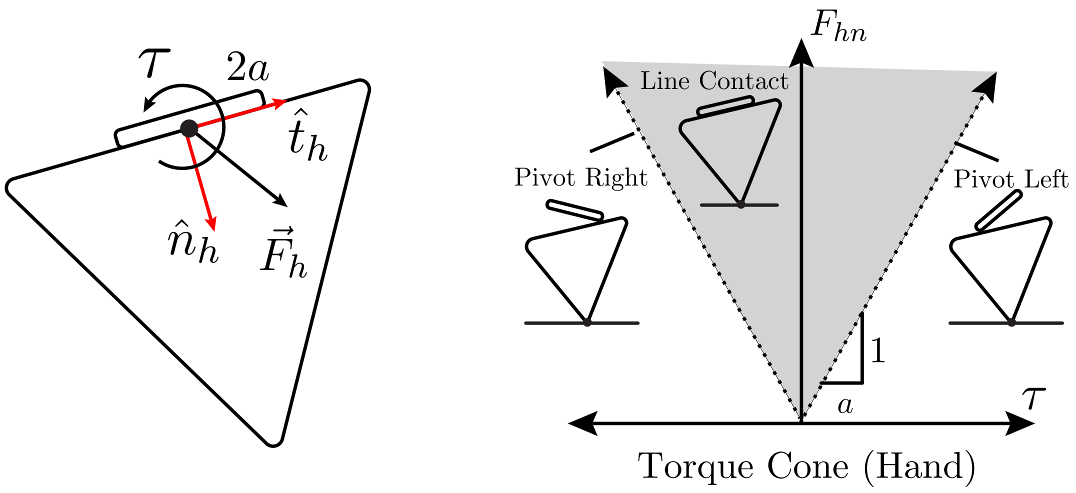


Figure D-5: Torque cone for the hand contact.

Let's now look at the torque constraints of the hand contact. We'll begin by assuming that the object face extends past the hand in both directions. In this case, we get the torque cone illustrated in Figure D-5:

$$-aF_{hn} \leq \tau \leq aF_{hn} \tag{D.23}$$

We can rewrite these two constraints as follows:

$$-aF_{hn} + \tau \leq 0 \quad \perp \quad \text{(hand pivot left)} \tag{D.24}$$

$$-aF_{hn} - \tau \leq 0 \quad \perp \quad \text{(hand pivot right)} \tag{D.25}$$

Substituting $w = [F_{hn}, F_{ht}, \tau]^T$ into the previous two inequalities, we get:

$$q_{lt1}^T w \leq b_{lt1} \quad \perp \quad \text{(hand pivot left)}, \qquad q_{lt1} = [-a, 0, 1]^T, \;\; b_{lt1} = 0 \tag{D.26}$$

$$q_{rt1}^T w \leq b_{rt1} \quad \perp \quad \text{(hand pivot right)}, \qquad q_{rt1} = [-a, 0, -1]^T, \;\; b_{rt1} = 0 \tag{D.27}$$

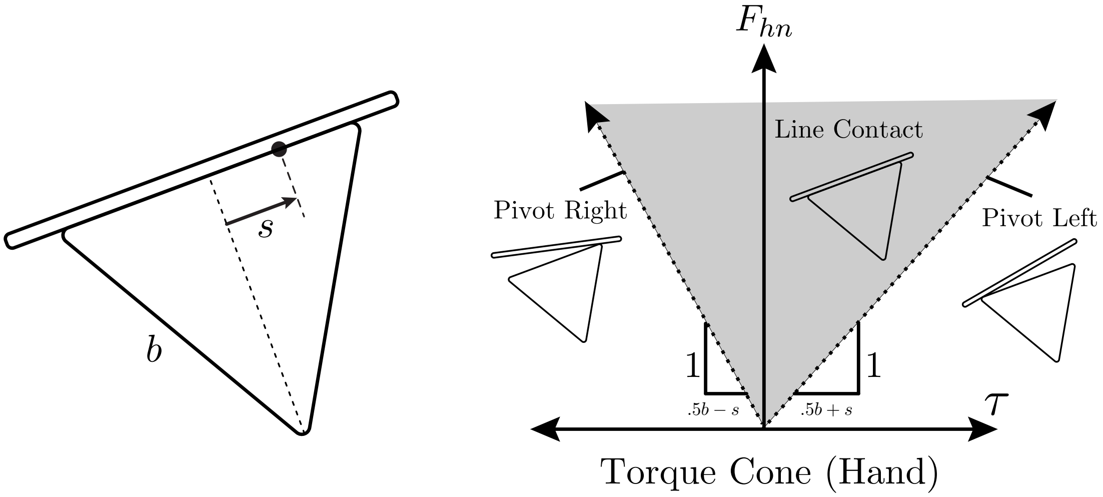


Figure D-6: Torque cone for the hand contact.

Now, we assume that the surface of the hand extends past the object contact surface in both directions. In this case, we get the torque cone illustrated in Figure D-6:

$$-(.5b - s)F_{hn} \leq \tau \leq (.5b + s)F_{hn} \tag{D.28}$$

We can rewrite these two constraints as follows:

$$-(.5b + s)F_{hn} + \tau \leq 0 \quad \perp \quad \text{(hand pivot left)} \tag{D.29}$$

$$-(.5b - s)F_{hn} - \tau \leq 0 \quad \perp \quad \text{(hand pivot right)} \tag{D.30}$$

Substituting $w = [F_{hn}, F_{ht}, \tau]^T$ into the previous two inequalities, we get:

$$q_{lt2}^T w \leq b_{lt2} \quad \perp \quad \text{(hand pivot left)}, \quad q_{lt2} = [-(.5b + s), 0, 1]^T, \;\; b_{lt2} = 0 \tag{D.31}$$

$$q_{rt2}^T w \leq b_{rt2} \quad \perp \quad \text{(hand pivot right)}, \quad q_{rt2} = [-(.5b - s), 0, -1]^T, \;\; b_{rt2} = 0 \tag{D.32}$$

Maintaining flush contact means enforcing all four of these torque inequalities (D.26,D.27,D.31,D.32). The last thing we need to consider is the torque balance equation: the net torque w/respect to the ground contact is zero:

$$\tau_A = 0 \tag{D.33}$$

$$mgl \sin\theta - sF_{hn} - dF_{ht} + \tau = 0 \tag{D.34}$$

$$\tag{D.35}$$

Substituting $w = [F_{hn}, F_{ht}, \tau]^T$ into this equation, we get:

$$q_e^T w = b_e \quad q_e = [-s, -d, 1]^T, \;\; b_e = -mgl \sin\theta \tag{D.36}$$

In summary, the equations we need to consider are as follows:

$$q_{rh}^T w \leq b_{rh} \quad \perp \quad \text{(hand slide right)}, \quad q_{rh} = [-\mu_h, 1, 0]^T, \;\; b_{rh} = 0 \tag{D.37}$$

$$q_{lh}^T w \leq b_{lh} \quad \perp \quad \text{(hand slide left)}, \quad q_{lh} = [-\mu_h, -1, 0]^T, \;\; b_{lh} = 0 \tag{D.38}$$

$$q_{rg}^T w \leq b_{rg} \quad \perp \quad \text{(object slide right)}, \quad q_{rg} = R^T(\theta) p_{rg}, \;\; b_{rg} = \mu_g mg \tag{D.39}$$

$$q_{lg}^T w \leq b_{lg} \quad \perp \quad \text{(object slide left)}, \quad q_{lg} = R^T(\theta) p_{lg}, \;\; b_{lg} = \mu_g mg \tag{D.40}$$

$$p_{rg} = \begin{bmatrix} -\mu_g \\ 1 \\ 0 \end{bmatrix}, \quad p_{lg} = \begin{bmatrix} -\mu_g \\ -1 \\ 0 \end{bmatrix}, \quad R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{D.41}$$

$$q_{lt1}^T w \leq b_{lt1} \quad \perp \quad \text{(hand pivot left)}, \qquad q_{lt1} = [-a, 0, 1]^T, \;\; b_{lt1} = 0 \tag{D.42}$$

$$q_{rt1}^T w \leq b_{rt1} \quad \perp \quad \text{(hand pivot right)}, \qquad q_{rt1} = [-a, 0, -1]^T, \;\; b_{rt1} = 0 \tag{D.43}$$

$$q_{lt2}^T w \leq b_{lt2} \quad \perp \quad \text{(hand pivot left)}, \qquad q_{lt2} = [-(.5b+s), 0, 1]^T, \;\; b_{lt2} = 0 \tag{D.44}$$

$$q_{rt2}^T w \leq b_{rt2} \quad \perp \quad \text{(hand pivot right)}, \qquad q_{rt2} = [-(.5b-s), 0, -1]^T, \;\; b_{rt2} = 0 \tag{D.45}$$

$$q_e^T w = b_e \qquad q_e = [-s, -d, 1]^T, \;\; b_e = -mgl\sin\theta \tag{D.46}$$

To check whether or not a motion is feasible for a given system configuration, we will search for a wrench that satisfies all of the wrench constraints, with the caveat that we require equality for the constraint that is complementary to the desired motion, and strict inequality for all other inequality constraints. For instance, to test for whether or not we can slide the hand to the right (relative to the object), we will search for a wrench that satisfies equations D.37-D.46, requiring equality at D.37, and strict inequality for all other inequality constraints.

We use this method to test the feasibility of sliding the hand left or right, sliding the object left or right, and maintaining static equilibrium, across a variety of parameter/variable sets. We assume an equilateral triangle with a side length of 13 cm and a mass of .22 kg. We assume that the end-effector has a length of 10 cm. We test feasibility for three pairs of friction coefficients: $(\mu_h = .1, \mu_g = .5)$, $(\mu_h = .3, \mu_g = .3)$, and $(\mu_h = .5, \mu_g = .1)$. Our simulation results are shown in Figures D-7 and D-8, in which we plot the regions of (θ, s) for which a given motion is feasible.

This type of feasibility analysis could potentially be used by a planner to determine which motion primitives are possible for a given state of the system.

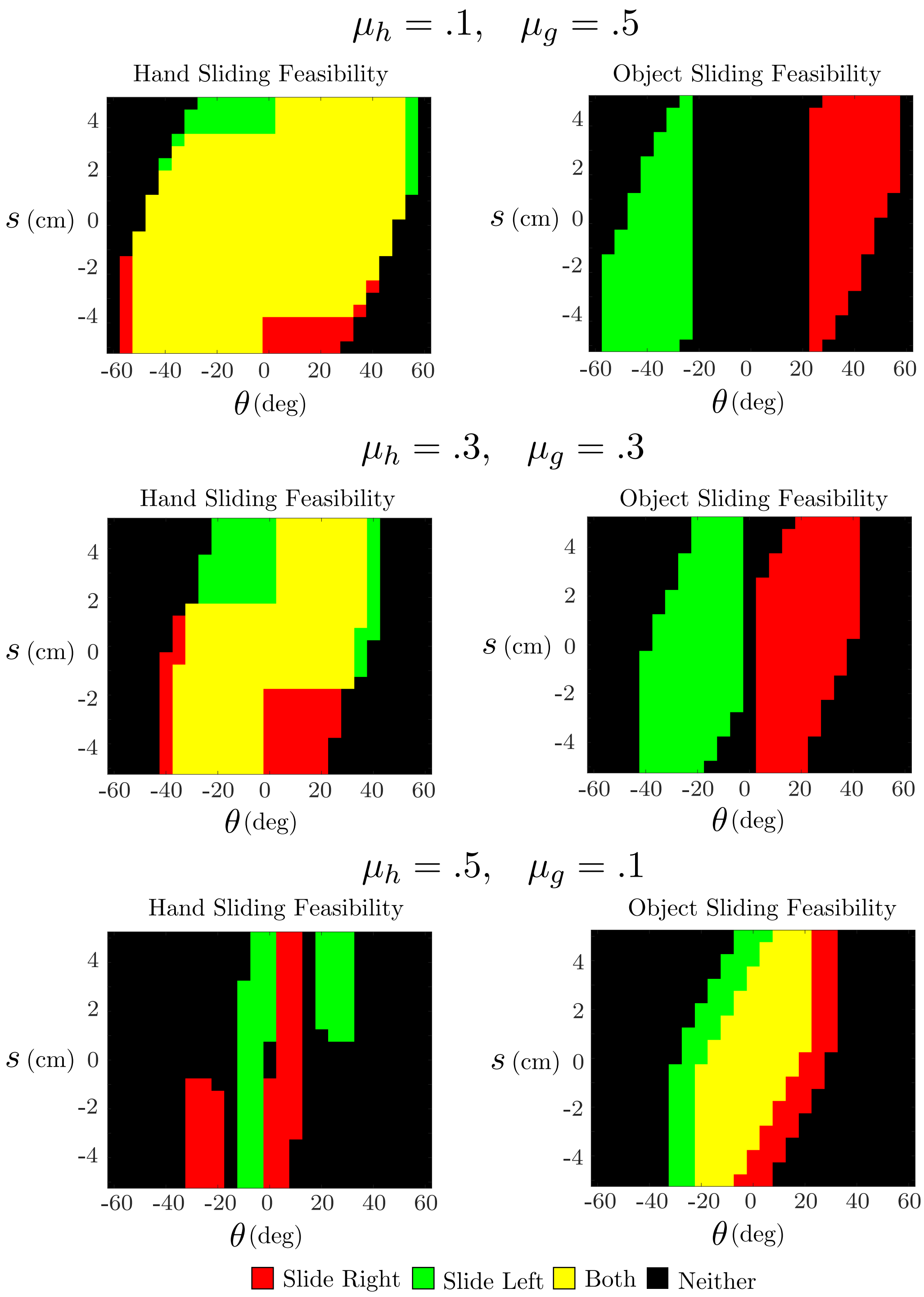
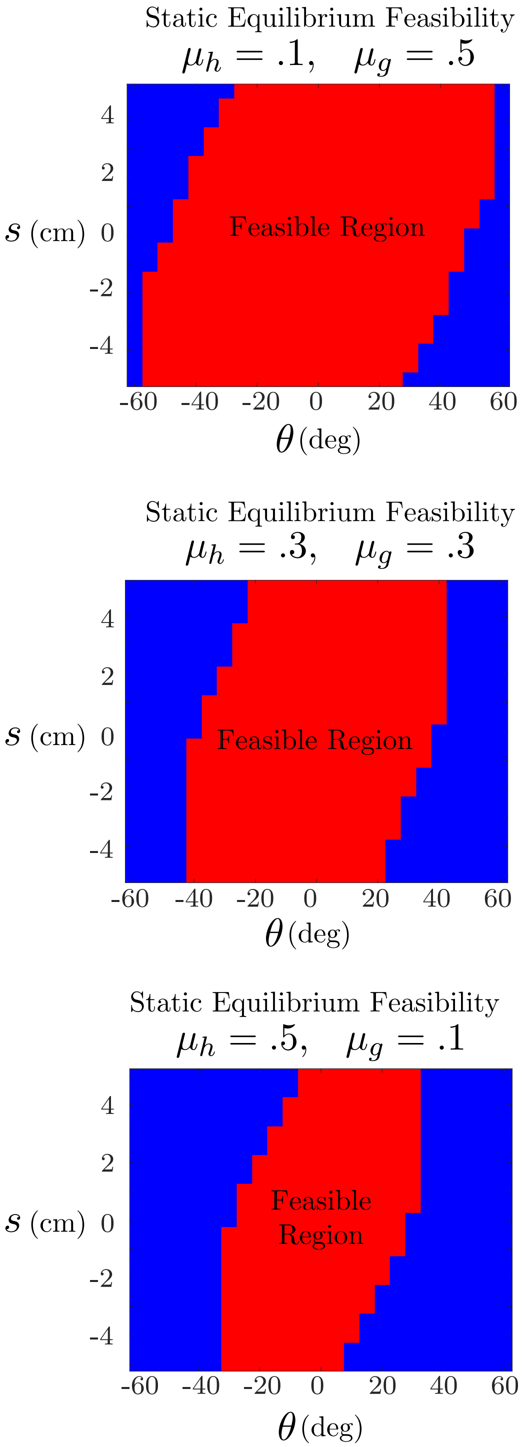


Figure D-7: State space regions for which the robot can slide the hand left/right or slide the object left/right.



Static Equilibrium Feasibility $\mu_h = .1, \quad \mu_g = .5$

Static Equilibrium Feasibility $\mu_h = .3, \quad \mu_g = .3$

Static Equilibrium Feasibility $\mu_h = .5, \quad \mu_g = .1$

Figure D-8: State space region for which the robot can maintain static equilibrium.

Bibliography

[1] Yasumichi Aiyama, Masayuki Inaba, and Hirochika Inoue. Pivoting: A new method of graspless manipulation of object by robot fingers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 136–143, 1993. 29

[2] Robert J. Anderson and Mark W. Spong. Hybrid impedance control of robotic manipulators. *IEEE Journal on Robotics and Automation*, 4(5):549–556, 1988. 32

[3] Devin J. Balkcom and Jeffrey C. Trinkle. Computing wrench cones for planar rigid body contact tasks. *The International Journal of Robotics Research*, 21(12):1053–1066, 2002. 30, 54

[4] Antonio Bicchi, J. Kenneth Salisbury, and David L. Brock. Contact sensing from force measurements. *The International Journal of Robotics Research*, 12(3):249–262, 1993. 31

[5] Joao Bimbo, Lakmal D. Seneviratne, Kaspar Althoefer, and Hongbin Liu. Combining touch and vision for the estimation of an object's pose during manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4021–4026. IEEE, 2013. 31

[6] Nikhil Chavan-Dafle, Rachel Holladay, and Alberto Rodriguez. Planar in-hand manipulation via motion cones. *The International Journal of Robotics Research*, 39 (2-3):163–182, 2020. 24, 29

[7] Xianyi Cheng, Eric Huang, Yifan Hou, and Matthew T. Mason. Contact mode guided

sampling-based planning for quasistatic dexterous manipulation in 2d. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 6520–6526. IEEE, 2021. 28

[8] Stefano Chiaverini and Lorenzo Sciavicco. The parallel approach to force/position control of robotic manipulators. *IEEE Transactions on Robotics and Automation*, 9 (4):361–373, 1993. 32

[9] Nikhil Chavan Dafle, Alberto Rodriguez, Robert Paolini, Bowei Tang, Siddhartha S. Srinivasa, Michael Erdmann, Matthew T. Mason, Ivan Lundberg, Harald Staab, and Thomas Fuhlbrigge. Extrinsic dexterity: In-hand manipulation with external forces. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1578–1585. IEEE, 2014. 44

[10] Joris De Schutter, Tinne De Laet, Johan Rutgeerts, Wilm Decré, Ruben Smits, Erwin Aertbeliën, Kasper Claes, and Herman Bruyninckx. Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *The International Journal of Robotics Research*, 26(5):433–455, 2007. 33

[11] Frank Dellaert and GTSAM Contributors. borglab/gtsam, May 2022. URL https://github.com/borglab/gtsam. 31, 87

[12] Frank Dellaert and Michael Kaess. Factor graphs for robot perception. *Foundations and Trends®*, 6(1-2):1–139, 2017. 143

[13] Kevin J. Doherty, David P. Baxter, Edward Schneeweiss, and John J. Leonard. Probabilistic data association via mixture models for robust semantic slam. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1098–1104. IEEE, 2020. 28

[14] Siyuan Dong, Devesh K. Jha, Diego Romeres, Sangwoon Kim, Daniel Nikovski, and Alberto Rodriguez. Tactile-rl for insertion: Generalization to objects of unknown

geometry. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2021. 33

[15] Neel Doshi, Orion Taylor, and Alberto Rodriguez. Manipulation of unknown objects via contact configuration regulation. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 2693–2699, 2022 ©2022 IEEE. 37

[16] Clemens Eppner and Oliver Brock. Planning grasp strategies that exploit environmental constraints. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 4947–4952. IEEE, 2015. 29

[17] Michael Erdmann. On a representation of friction in configuration space. *The International Journal of Robotics Research*, 13(3):240–271, 1994. 30, 54

[18] Takuto Fujiki and Kenji Tahara. Series admittance–impedance controller for more robust and stable extension of force control. *ROBOMECH*, 9(1):23, 2022. 32

[19] T. Furukawa, D.C. Rye, M.W.M.G. Dissanayake, and A.J. Barratt. Automated polishing of an unknown three-dimensional surface. *Robotics and Computer-Integrated Manufacturing*, 12(3):261–270, 1996. 33

[20] Suresh Goyal, Andy Ruina, and Jim Papadopoulos. Planar sliding with dry friction part 1. limit surface and moment function. *Wear*, 143(2):307–330, 1991. 30

[21] Ross Hartley, Maani Ghaffari Jadidi, Lu Gan, Jiunn-Kai Huang, Jessy W. Grizzle, and Ryan M. Eustice. Hybrid contact preintegration for visual-inertial-contact state estimation using factor graphs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3783–3790. IEEE, 2018. 31

[22] Ross Hartley, Josh Mangelson, Lu Gan, Maani Ghaffari Jadidi, Jeffrey M. Walls, Ryan M. Eustice, and Jessy W. Grizzle. Legged robot state-estimation through combined forward kinematic and preintegrated contact factors. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4422–4429. IEEE, 2018. 31

[23] Paul Hebert, Nicolas Hudson, Jeremy Ma, and Joel Burdick. Fusion of stereo vision, force-torque, and joint sensors for estimation of in-hand object location. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 5935–5941. IEEE, 2011. 31

[24] M. G. Her and H. Kazerooni. Automated robotic deburring of parts using compliance control. *Journal of Dynamic Systems, Measurement, and Control*, 113(1):60–66, 03 1991. 33

[25] Francois R Hogan and Alberto Rodriguez. Reactive planar non-prehensile manipulation with hybrid model predictive control. *The International Journal of Robotics Research*, 39(7):755–773, 2020. 24, 27, 28

[26] Francois R. Hogan, Jose Ballester, Siyuan Dong, and Alberto Rodriguez. Tactile dexterity: Manipulation primitives with tactile feedback. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2020. 28, 33

[27] Neville Hogan. Impedance control: An approach to manipulation: Parts i–iii. *ASME J. Dyn. Syst. Meas. Control*, 107:1–24, 1985. 32

[28] Anne Holladay, Robert Paolini, and Matthew T. Mason. A general framework for open-loop pivoting. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 3675–3681. IEEE, 2015. 29

[29] Yifan Hou and Matthew T. Mason. Criteria for maintaining desired contacts for quasi-static systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6555–6561, 2019. 33, 105

[30] Yifan Hou and Matthew T. Mason. Robust execution of contact-rich motion plans by hybrid force-velocity control. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1933–1939. IEEE, 2019. 32, 33, 105

[31] Yifan Hou, Zhenzhong Jia, and Matthew Mason. Manipulation with Shared Grasping. In *Proceedings of Robotics: Science and Systems*, Corvalis, Oregon, USA, July 2020. doi: 10.15607/RSS.2020.XVI.086. 33, 105

[32] Ming Hsiao and Michael Kaess. Mh-isam2: Multi-hypothesis isam using bayes tree and hypo-tree. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1274–1280. IEEE, 2019. 28

[33] Gregory Izatt, Geronimo Mirano, Edward Adelson, and Russ Tedrake. Tracking objects with point clouds from vision and touch. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 4000–4007. IEEE, 2017. 31

[34] Nawid Jamali, Carlo Ciliberto, Lorenzo Rosasco, and Lorenzo Natale. Active perception: Building objects' models using tactile exploration. In *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 179–185, 2016. 31

[35] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J. Leonard, and Frank Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012. 31, 87

[36] Imin Kao and Mark R. Cutkosky. Quasistatic manipulation with compliance and sliding. *The International Journal of Robotics Research*, 11(1):20–40, 1992. 30, 33

[37] Yiannis Karayiannidis, Christian Smith, Francisco E Vina, Petter Ögren, and Danica Kragic. Model-free robot manipulation of doors and drawers by means of fixed-grasps. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 4485–4492, 2013. 33

[38] Sangwoon Kim and Alberto Rodriguez. Active extrinsic contact sensing: Application to general peg-in-hole insertion. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 10241–10247. IEEE, 2022. 29

[39] Michael C. Koval, Nancy S. Pollard, and Siddhartha S. Srinivasa. Pose estimation for planar contact manipulation with manifold particle filters. *The International Journal of Robotics Research*, 34(7):922–945, 2015. 31

[40] Tine Lefebvre, Herman Bruyninckx, and Joris De Schutter. Polyhedral contact formation modeling and identification for autonomous compliant motion. *IEEE Trans-*

actions on Robotics and Automation, 19(1):26–41, 2003. doi: 10.1109/TRA.2002.805677. 33

[41] Shuai Li, Siwei Lyu, and Jeff Trinkle. State estimation for dynamic systems with intermittent contact. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 3709–3715. IEEE, 2015. 31

[42] Jacky Liang, Xianyi Cheng, and Oliver Kroemer. Learning preconditions of hybrid force-velocity controllers for contact-rich manipulation. In *arXiv preprint*, 2022. 33, 116

[43] Kevin M. Lynch. Estimating the friction parameters of pushed objects. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1993. 30

[44] Kevin M. Lynch and Matthew T. Mason. Dynamic underactuated nonprehensile manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 889–896, 1996. 27

[45] Kevin M. Lynch and Matthew T. Mason. Dynamic nonprehensile manipulation: Controllability, planning, and experiments. *The International Journal of Robotics Research*, 18(1):64–92, 1999. 27

[46] Kevin M. Lynch, Hitoshi Maekawa, and Kazuo Tanie. Manipulation and active sensing by pushing using tactile feedback. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1992. 30

[47] Daolin Ma, Siyuan Dong, and Alberto Rodriguez. Extrinsic contact sensing with relative-motion tracking from distributed tactile measurements. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 11262–11268. IEEE, 2021. 29, 33

[48] William D. MacMillan. Dynamics of rigid bodies. 64, 1960. 30

[49] Lucas Manuelli and Russ Tedrake. Localizing external contact using proprioceptive sensors: The contact particle filter. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016. 31

[50] Matthew T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6):418–432, 1981. 32

[51] Matthew T. Mason. Mechanics and planning of manipulator pushing operations. *The International Journal of Robotics Research*, 5(3):53–71, 1986. 30

[52] Wim Meeussen, Johan Rutgeerts, Klaas Gadeyne, Herman Bruyninckx, and Joris De Schutter. Particle filters for hybrid event sensor fusion with 3d vision and force. In *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 518–523. IEEE, 2006. 31

[53] Claudio Melchiorri. Slip detection and control using tactile and force sensors. *IEEE/ASME transactions on mechatronics*, 5(3):235–243, 2000. 30

[54] Michael Montemerlo and Sebastian Thrun. Simultaneous localization and mapping with unknown data association using fastslam. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1985–1991. IEEE, 2003. 28

[55] Günter Niemeyer and J-JE Slotine. A simple strategy for opening an unknown door. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1448–1453, 1997. 33

[56] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407, 2011. doi: 10.1109/ICRA.2011.5979561. 108

[57] Anna Petrovskaya and Oussama Khatib. Global localization of objects via touch. *IEEE Transactions on Robotics*, 27(3):569–585, 2011. 31

[58] Marc H. Raibert and John J. Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103(2):126–133, 06 1981. 32

[59] J Kenneth Salisbury. Active stiffness control of a manipulator in cartesian coordinates. In *IEEE conference on decision and control*, 1980. 32

[60] Brad Saund, Shiyuan Chen, and Reid Simmons. Touch based localization of parts for high precision manufacturing. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 378–385. IEEE, 2017. 31

[61] Yu She, Shaoxiong Wang, Siyuan Dong, Neha Sunil, Alberto Rodriguez, and Edward Adelson. Cable manipulation with a tactile-reactive gripper. *The International Journal of Robotics Research*, 2021. doi: 10.1177/02783649211027233. 24, 33

[62] Yuki Shirai, Devesh K. Jha, Arvind U. Raghunathan, and Dennis Hong. Tactile tool manipulation. In *arXiv preprint*, 2023. 33

[63] Andrea Sipos and Nima Fazeli. Simultaneous contact location and object pose estimation using proprioception and tactile feedback. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3233–3240, 2022. 31

[64] Claudius Strub, Florentin Wörgötter, Helge Ritter, and Yulia Sandamirskaya. Using haptics to extract object shape from rotational manipulations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2179–2186. IEEE, 2014. 31

[65] Sudharshan Suresh, Maria Bauza, Kuan-Ting Yu, Joshua G. Mangelson, Alberto Rodriguez, and Michael Kaess. Tactile slam: Real-time inference of shape and pose from planar pushing. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 11322–11328. IEEE, 2021. 31, 86

[66] Filipe Veiga, Herke Van Hoof, Jan Peters, and Tucker Hermans. Stabilizing novel objects by learning to predict tactile slip. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5065–5072, 2015. 30

[67] Sean Wang, Ankit Bhatia, Matthew T. Mason, and Aaron M. Johnson. Contact localization using velocity constraints. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020. 31

[68] Harry West and Haruhiko Asada. A method for the design of hybrid position/force controllers for manipulators constrained by contact with the environment. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 251–259. IEEE, 1985. 32

[69] J. Zachary Woodruff and Kevin M. Lynch. Planning and control for dynamic, nonprehensile, and hybrid manipulation tasks. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 4066–4073. IEEE, 2017. 27

[70] Tsuneo Yoshikawa and Masamitsu Kurisu. Identification of the center of friction from pushing an object by a mobile robot. In *Proceedings of IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS)*, 1991. 30

[71] Kuan–Ting Yu and Alberto Rodriguez. Realtime state estimation with tactile and visual sensing. application to planar manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 7778–7785. IEEE, 2018. 31

[72] Jiaji Zhou, Matthew T Mason, Robert Paolini, and Drew Bagnell. A convex polynomial model for planar sliding mechanics: theory, application, and experimental validation. *The International Journal of Robotics Research*, 37(2-3):249–265, 2018. 30