NORTHWESTERN UNIVERSITY

Modeling and Motion Planning for In-hand Sliding Manipulation

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Mechanical Engineering

By

Jian Shi

EVANSTON, ILLINOIS

June 2018

ABSTRACT

Modeling and Motion Planning for In-hand Sliding Manipulation

Jian Shi

This thesis studies the in-hand manipulation problem of repositioning finger contacts on an object by controlled sliding. In this thesis we investigate two versions of the problem. First for a multifingered hand with circle patch contacts, we present a framework for planning the motion of the hand to create an inertial load on the grasped object to achieve a desired in-grasp sliding motion. The model of the sliding dynamics is based on a soft-finger limit surface contact model at each fingertip. A motion planner is derived to automatically solve for the finger motions for a given initial and desired configuration of the object relative to the fingers. Iterative planning and execution is shown to reduce errors that occur due to modeling and trajectory tracking errors. The framework is applied to the problem of regrasping a laminar object held in a pinch grasp. We propose a limit surface model of the contact pressure distribution at each finger to predict sliding directions. Experimental validations are shown, including iterative error reduction and repeatability of the experiment.

Secondly we study quasistatic in-hand sliding manipulation with spring-sliding compliant grasps. We focus on point-contact multi-fingered grasps and the goal is to achieve object regrasping by taking advantage of external contacts with the environment. Spring

compliance ensures fingers remain in contact and maps contact forces to finger compressions. By controlling finger anchor motions the contact forces can be moved to the edges of friction cones and cause sliding to realize regrasping. External contacts provide forces that maintain object force balance during the motions. We model the contact and object mechanics for multi-fingered grasps in spatial cases and analyze robust conditions in terms of finger contact wrench uncertainties. Based on the modeling a general motion planning framework is proposed. We use a two-fingered system to illustrate the analysis and detail the planning algorithm to find feasible regrasp motions maximizing robustness.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

Most human, animal, and even robot manipulation tasks involve controlling motion of the object relative to the manipulator, particularly in nonprehensile (graspless) manipulation modes such as pushing, rolling, pivoting, tipping, tapping, and kicking. Even in pick-carry-place manipulation, where the carry portion of the task keeps the object stationary relative to the hand, the pick and place phases typically involve the object sliding or rolling on the fingers as the hand achieves a firm grasp or lets the object go. Other examples of controlled relative motion in grasping manipulation include finger gaiting, where the fingers quasistatically “walk” over the object to achieve a regrasp, all the while maintaining a stable grasp; rolling the object on the fingertips; and letting the object slide relative to the fingertips. Together these may be referred to as in-hand manipulation.

In-hand manipulation can increase the dexterity of a hand/gripper. For example, when picking up a pen, you can adjust your grasp by pushing your pen against your fingers. More aggressively, if you relax your grip and move your hand, the pen can slip between the fingers. These motions can refine the grasp after the pen was initially picked up.

In this thesis, we study repositioning of finger contacts relative to the object by controlled sliding. By defining a grasp configuration as a set of all finger contact positions relative to the object, the main problem of this thesis is to find a hand motion that can regrasp an object to a desired new grasp configuration. The following aspects of the hand-object system should be considered in mechanical modeling:

- *Finger contact types.* Point, line or patch contacts.
- *External force sources.* Besides forces from finger contacts, other forces applied to the object may come from contacts with environment, gravity, inertia forces, etc.
- *Compliance.* In this thesis compliance refers to yielding behaviors in robot fingers that ensure bounded fingertip contact forces, such as spring, damping, and mass compliances. Following this definition, sliding can be treated as another source of compliance that bound finger tangential contact forces relative to normal forces.

Based on the taxonomy, we investigate in-hand sliding manipulation in two ways:

1) Regrasping using dynamic loads on the object: The hand uses object inertia to cause sliding to the desired new grasp by accelerating the hand beyond the point the finger friction forces can resist relative motion of the object. A framework is presented for planning the motion of an n-fingered robot hand to create an inertial load on a grasped object to achieve a desired in-grasp sliding motion. The model of the sliding dynamics is based on a soft-finger limit surface contact model at each fingertip. A motion planner is derived to automatically solve for the finger motions for a given initial and desired configuration of the object relative to the fingers. The framework is applied to the problem of regrasping a laminar object held in a pinch grasp. Iterative planning and execution are shown to reduce the errors that occur due to the modeling and trajectory tracking errors [**57, 58**].

2) Regrasping using external contacts with spring-sliding compliance: In the previous approach, an object is regrasped using dynamic loads and sliding is restricted to parallel surfaces. To enable sliding on general surfaces, we incorporate spring compliance of the fingers. Spring compliance maps contact forces to finger compressions and maintains the contact between the fingers and the object. This analysis focus on point-finger contacts and quasistatic mechanics. Each finger is modeled as a frictional point contact connected by a three-dimensional linear spring to an anchor point whose motions is controlled in three linear directions. By controlling finger anchor motions the contact forces can be controlled due to the springs and sliding regrasps occur when the contact forces are at the edges of their friction cones. External contacts provide forces that maintain object force balance during the motions. We model the contact and object mechanics for multi-fingered grasps in spatial cases and analyze robust conditions in terms of finger contact wrench uncertainties. We propose a general motion planning framework based on the modeling. We use a two-fingered system to illustrate the analysis and detail the planning algorithm to find feasible regrasp motions maximizing robustness. The approach is validated in both simulation and experiment with a planar regrasping example.

1.1. Thesis Outline

The rest of this thesis is arranged as follows: Chapter 2 describes related works. Chapter 3 introduces and describes the development of the experimental environment, the ERIN system. Chapter 4 describes regrasping using dynamic loads. The content of this chapter has appeared in [**57, 58**]. Chapter 5 presents modeling and motion planning of in-hand manipulation with spring-sliding compliance. Chapter 6 gives the conclusion and a discussion on limitations and future directions for this work.

CHAPTER 2

Related Work

2.1. In-hand Manipulation

There has been extensive work on kinematic in-hand manipulation where an object is moved relative to a finger without breaking contact or sliding on the surface. This is sometimes referred to as "precision manipulation." Salisbury et al. [**33, 52, 51, 42**] formulated kinematics of multi-fingered hands for several different types of frictional finger contacts. Dexterous manipulation relies on controlling fingertip contact forces applied to the object. Li, Hsu, and Sastry [**37**] and Yoshikawa and Nagai [**69**] use rigid, rolling finger contacts to calculate grasp stability, manipulability, and to develop controllers for tracking a position trajectory while maintaining a desired grasp force. Subsequent work on rolling manipulation includes [**36, 53, 4, 22**]. More recent work by Rojas and Dollar [**49**] has expanded this to estimate the precision manipulation capabilities of arbitrary manipulator/object configurations for use in autonomous manipulation planning. Sundaralingam and Hermans [**60**] uses kinematic trajectory optimization to find quasistatic finger motions to reposition an object relative to the palm.

Due to manipulator joint limits and workspace constraints, the set of reachable states using precision manipulation is generally small and the motions are slow. In-hand sliding can be used to quickly reposition an object in the hand without requiring large movements of the fingers. Brock [**6**] and Cole and Sastry [**12**] explored methods to reposition objects using controlled slip. Trinkle and Hunter [**61**] extended the dexterous manipulation planning problem to consider rolling and slipping contact modes. The hybrid planning problem was further developed by Yashima et al. [**68, 67**]. The space of reachable object states can be further expanded by breaking contact with a single finger, moving it, and regrasping the object while the remaining fingers keep the object in force closure. This in-hand regrasp technique is called finger gaiting [**16, 25, 10, 47, 70, 23, 56, 50, 11, 41**].

Dynamic forces can also be used for in-hand manipulation. Furukawa et al. [**17**] demonstrated regrasping by tossing a foam cylinder up and catching it. Arisumi et al. have explored the idea of casting manipulation where a manipulator is thrown and its "free flight" trajectory can be controlled in midair using tension forces in a tether [**1**]. Chavan-Dafle et al. [**9**] tested hand-coded regrasps that take advantage of external forces such as gravity, dynamic forces, and contact with the environment to regrasp objects using a simple manipulator. Chavan-Dafle and Rodriguez [**7**] explored in-hand manipulation of an object by external contacts with the environment. With designed finger actions, motions of the object were simulated and validated experimentally with different shapes of contacts. More recently in [**8**], they studied the mechanics of alternating sticking contact and proposed a sampling-based planning framework to concatenate feasible and stable pushes to achieve the desired regrasp. Feedback is used in [**35**] to adapt the grasp by increasing stiffness and repositioning fingers to improve the grasp quality. Viña et al. [**62, 63**] showed that by using adaptive control with vision and tactile feedback, monodirectional pivoting of an object pinched by a pair of fingers can be achieved by changing the gripping forces. Kumar et al. [**34**] programmed a pneumatically-actuated hand to learn in-hand manipulation skills using model-based reinforcement learning. Sintov and Shapiro [**59**] developed an algorithm to swing up a rod by generating gripper motions, in which the contact point was modeled as a pivot joint that can apply frictional torques. The method was validated in simulation. Hou et al. [**26**] studied dynamic planar pivoting of a pinched object driven by the hand swing motion and contact normal force control.

2.2. Friction Modeling

Friction provides tangential contact forces and it plays an important role in contact-rich manipulation. There is an extensive literature on how to model friction [5, 2, 46]. In this thesis friction is assumed to follow Coulomb's law [13, 40] with equal static and kinetic coefficients. For a point contact, the friction cone is a useful geometric interpretation of Coulomb's law [44]. For multiple contact points, the cone of possible reaction forces is simply the convex hull of the individual cones.

For patch contacts, Howe et al. [29] and Goyal et al. [19, 20, 21] describe the concept of a limit surface as a two-dimensional surface in a three-dimensional force-moment space with two tangential linear force directions and one moment about the contact normal. The limit surface defines the maximum set of external wrenches that can be resisted by the frictional forces due to the contact. Xydas and Kao [66] derived models of soft-finger contacts and the resulting limit surfaces. The ellipsoid limit surface model has been used in the problem of planar pushing [38, 39]. Recent work by Zhou et al. [71] proposed a fourth-order polynomial limit surface model for planar sliding and identified model parameters using simulation and experimental data.

2.3. Compliant Grasps

Hanafusa and Asada [24] first modeled the spring compliance with frictionless elastic fingers and formulated grasp stability. A stable grasp means that for small position disturbance of the object the grasp is able to restore the object to its desired location. The grasp stability is determined by finger stiffness and local contact geometry. Baker et al. [3] further developed the stability conditions with the same system. Expanded to general cases, Howard and Kumar [28] classified categories of equilibrium grasps and derived determinations of stability respectively. Cutkosky and Kao [14] achieved a grasp stiffness by controlling finger joint stiffness, and analyzed the properties of the grasp.

For a similar problem as addressed in this work, Cutkosky and Kao [32] modeled sliding manipulation with spring compliance and limit surfaces. The motion of the contact points were solved by assuming infinitesimal motions while the magnitude of the sliding velocity is fixed. In our work we assume finite sliding velocity and use the constraint that when sliding the finger contact force remain on the boundary of the friction cone to solve the velocity magnitude of the fingertip sliding. More recent work by Odhner and Dollar [45] showed in-hand rolling with an underactuated compliant hand.

Compliant grasps have applications in assembly. The remote center of compliance (RCC) device is a mechanical solution to reduce mating forces and the chance of jamming in certain assembly operations [64, 15, 65]. Peshkin et al. [18] generalized the idea by outlining a design strategy for passive devices to implement desired spring characteristics. Schimmels et al. [54, 55] derived conditions for accommodation control to yield error-corrective assembly with frictional contacts. Ji and Xiao [31] explored methods to plan compliant assembly based on a contact state graph. Meeussen et al. [43] developed an approach to convert a contact path into a force-based task specification for executing the compliant path via hybrid position and force control. Park et al. [48] developed a procedure and a controller that yield compliant behavior without force feedback nor passive compliance mechanisms to solve the peg-in-hole assembly problem.

CHAPTER 3

Experimental Environment Development

This chapter introduces the hardware platform used for all the experiments conducted in this thesis. We first give a description of the hardware and software architecture of the system and then show the algorithm of the vision data filtering. In the end a task of soft-catching a ball is shown to demonstrate dynamic manipulation capability of the system.

3.1. The ERIN Instrumented Manipulation Environment

As the experiment platform of this thesis, the ERIN Instrumented Manipulation Environment consists of a 7-DOF WAM arm with high-resolution encoders and a 6-DOF wrist force-torque sensor (WAM refers to Whole Arm Manipulator and is designed and manufactured by *Barrett Technology*® *Inc.*); an Allegro 16-dof four-fingered robot hand; four Syntouch biotac tactile sensor fingertips; and a ten-camera 360 Hz ***NaturalPoint***® *OptiTrack*™ IR-based motion capture system as shown in Fig. 3.1.

This system is designed to implement experiments in dexterous manipulation, dynamic manipulation, and human-robot interaction. It supports the experimental implementations of the research topics described in this thesis. The high maneuverability of the robot arm allows the manipulator to generate dynamic loads on objects in a grasp, and these loads can be used to initiate in-hand sliding. The high-speed tracking system allows 3D gross motion tracking of objects in a sliding grasp. The dexterity of the robot hand allows a wide variety of grasp configurations to be achieved, therefore allowing control over the compliant sliding properties. The goal of the tactile sensors is to allow fine control

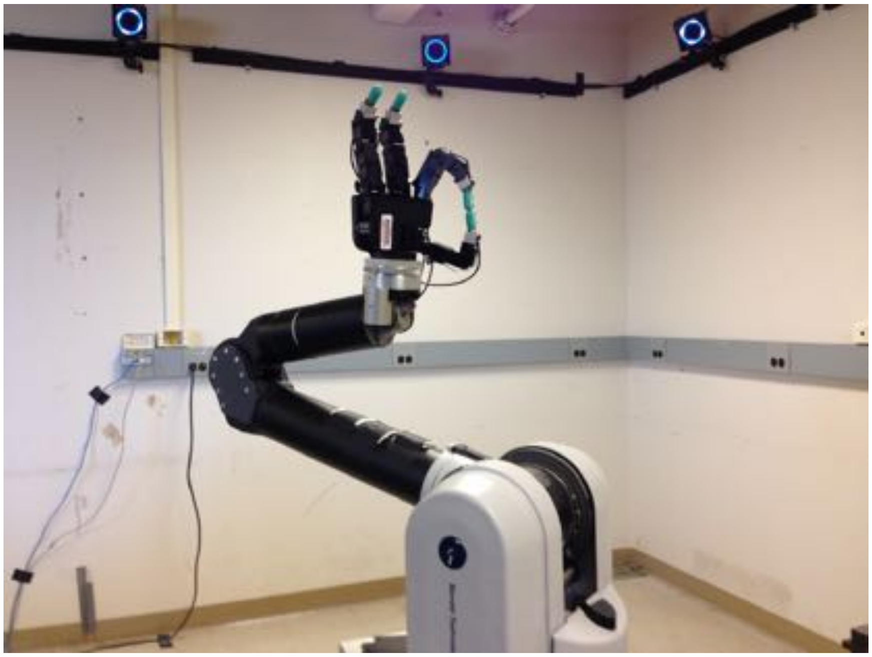


Figure 3.1. The ERIN instrumented manipulation environment, showing the WAM arm, the Allegro hand, four Syntouch biotac sensor fingertips, and part of the 10-camera *OptiTrack*™ motion capture system.

of grasp contact forces. The Syntouch biotac sensors are proved to be hard to achieve this goal through a serious of experiments. Main limitations are difficulties to reason signals from impedance sensing electrodes to contact forces and signal drifting effect due to environment temperature changes. We have also tested *Optoforce* sensors which have advantages of much more reliable and accurate contact force sensing but with limitations of low respond bandwidth from the compliant material of the contact dome, and the designed sensing contact surface need to be at the top of the contact dome.

For the experiments conducted in this thesis, none of the tactile and force sensors were used. Therefore for the rest of this chapter, we focus on integrating the WAM arm, the Allegro hand, and the high-speed vision system. And for the rest of this thesis the ERIN system refers to combination of these three.

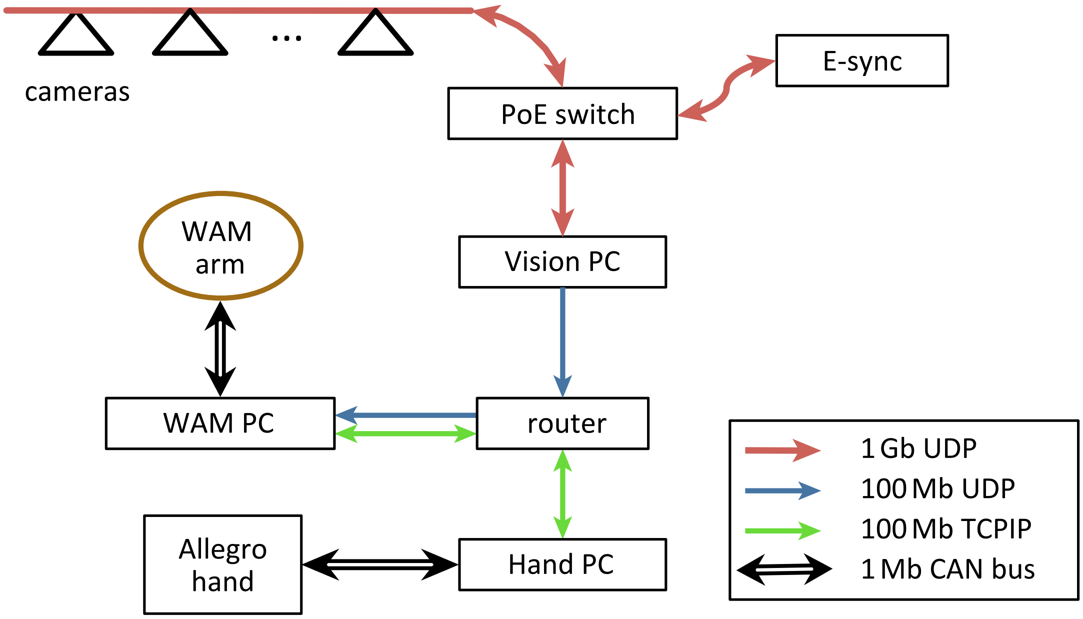


Figure 3.2. Hardware architecture of the WAM and vision system.

3.2. System Architecture

In this section we introduce the design of the system architecture in hardware and software levels. Since the subsystems are from different manufactures and running at different rates, building the communication between them is the primary task.

3.2.1. Hardware Architecture

The design of the hardware architecture is shown in Fig. 3.2. To increase modularity and computing power of the system, we use multiple PCs to control the hardwares:

- The WAM arm is controlled by an external computer named WAM PC, which has an Intel® Core™ i7-4770 CPU and 16 GB RAM. The WAM PC sends command motor torques to *Pucks*, which are the motor controller of the WAM and read encoders data from it through a 1 Mb CAN bus.

- The allegro hand is controlled by the Hand PC, which has an Intel® Core™ i7-3770k CPU and 8 GB RAM. Information of commanded torques and joint encoder readings are transmitted through another 1 Mb CAN bus.
- For the *OptiTrack*™ motion capture system, the Vision PC has an Intel® Core Quad Q8400 CPU and 3 GB RAM. Camera data are sent to the Vision PC through a standalone 1 Gb network.

The WAM PC and Hand PC are running Linux and the Vision PC is a Windows machine. The choices of the PCs are based on the used APIs which will be detailed in the software architecture section. All of the PCs are connected in a local network.

For more details of the motion capture system, the *OptiTrack*™ system consists of 10 cameras, a PoE switch, and an E-Sync device. The cameras are infrared cameras with high capturing rate up to 360 Hz and 1.7 MP resolution (1664 × 1088). Balls covered with reflective paint are used as markers that reflect infrared lights from LEDs mounted on the cameras. The PoE (Power over Ethernet) switch transmits data between devices and also powering the cameras and the E-Sync. The E-Sync is sending control and triggering signals to the cameras which is controlled by the Vision PC. By default the E-Sync generates periodical signals to trigger the cameras and the frequency of the signals are controlled by the program running at the Vision PC. Since there is no clock in the E-Sync and the Vision PC only sends reference frequency signals, the captured frames are not time-stamped by default, which means the exact time when each frame was captured is unknown. Although there is an alternative method by using external SMPTE signals to trigger the cameras and time-stamping the frames, it is limited by the maximum of 30 Hz update rate from the SMPTE protocol. It turns out that **there is no clean way to accurately time-stamp camera frames** using the triggering mechanism when the cameras are running at high-speed.

An experiment was implemented to estimate the delay from when a camera frame is captured to the time when corresponding vision data is received by the WAM PC. The camera control program named **Motive** measures the time delay from the capture time to the time when 3D-reconstructed vision data is ready. The mean value of this delay is 0.2 ms with a standard deviation of 0.05 ms. After synchronizing clocks of WAM PC and Vision PC, we measured the delay of data transmission between them: mean value of the delay is 0.065 ms with a standard derivation of 1.3 ms. With low occurrence, some notable delays can reach up to around 20 ms compared to the control loop period (2 ms at 500 Hz).

3.2.2. Software Architecture

Fig. 3.3 shows the software architecture of the ERIN system. The WAM control programs run at the WAM PC based on `libbarrett`, a C++ library written and maintained by *Barrett Technology, Inc.* including a low-level and a high-level interfaces for controlling the WAM [1]. The library generates a realtime thread to compute the controls of the WAM under a version of Linux kernel modified by `Xenomai`.

The Hand PC runs ROS indigo under Ubuntu 14.04 to control the allegro hand. The software is built upon Allegro Hand ROS stack[2] which contains low-level communications with the hand and joint torque and position controllers. We add several functions to the *allegro_hand* node to calculate finger endpoint positions and create a topics to publish the information. We also add another node called *sockets* to communicate with the WAM PC. The fingertip positions are calculated and published to a buffer at 333 Hz. The *sockets* node receives control commands from the WAM PC and sends latest fingertip positions in the buffer upon request.

[1] More details can be found at `http://web.barrett.com/libbarrett/`.

[2] `https://github.com/felixduvallet/allegro-hand-ros`.

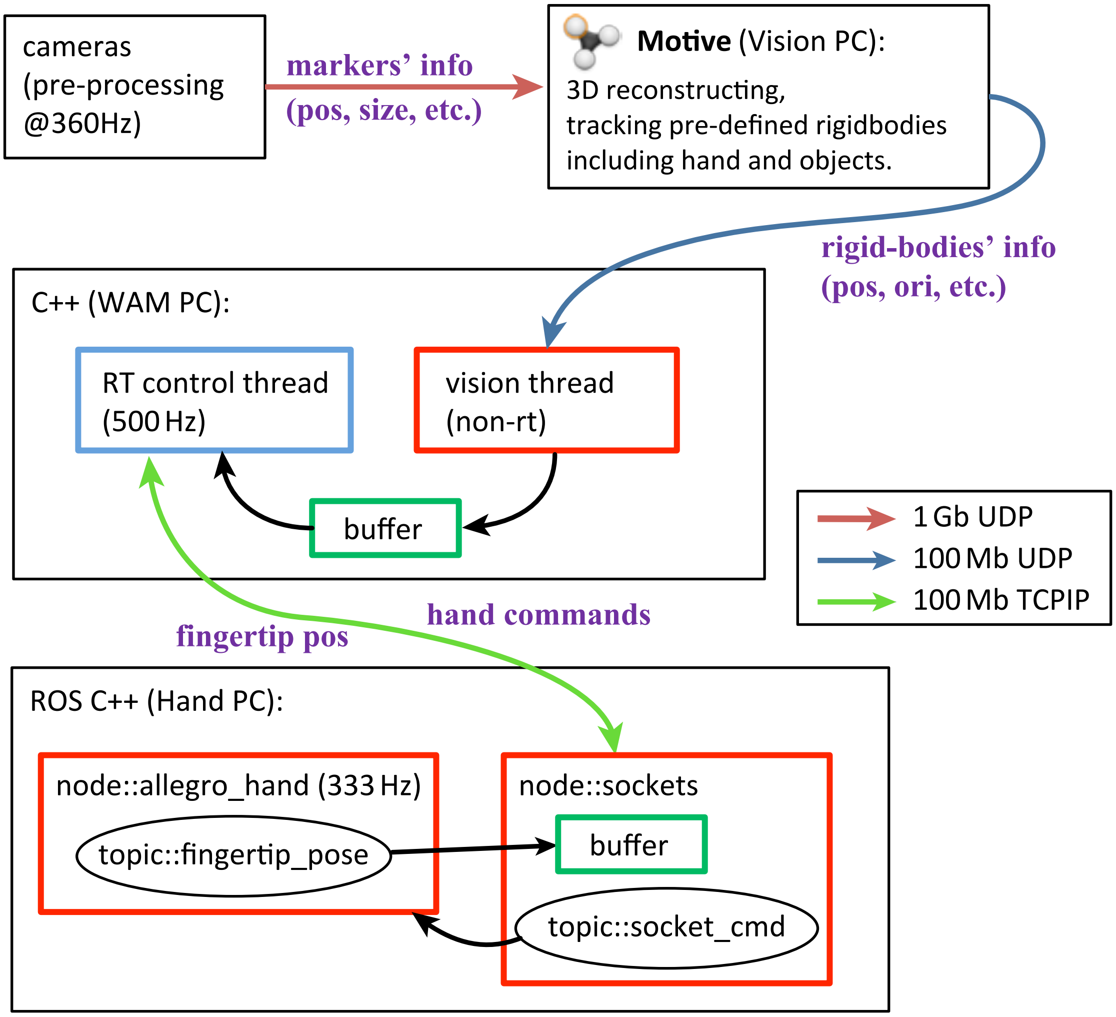


Figure 3.3. Software architecture of the ERIN system.

For the vision data, after camera frames are captured, raw images are first pre-processed by on-camera micro chips. On each camera after a frame is captured, all marker lumps will be detected and then calculate and send information including 2D location, size and roundness of the lumps to **Motive** software running at the Vision PC. **Motive** receives and filters the 2D marker data and then reconstructs 3D positions of the markers. To track 3D rigid-bodies pose information (center positions and orientations), users can use multiple markers (at least 3) to define polyhedron-shaped rigid-bodies that are fixed to the tracking objects. Then when a new camera frame data is received and filtered, the program will try to fit subsets of the 3D markers locations to the pre-defined polyhedrons, and if any fits are found, the locations and orientations of the frames attached to the polyhedrons will be updated.

Since **Motive** can only run under Windows platform and the WAM PC is running Linux, we set the **Motive** to publish processed rigid-bodies information through UDP and the data is received and decoded in the WAM PC. *NaturalPoint*® supplies NatNet SDK, a Client/Server networking SDK for sending and receiving vision data across networks. In the WAM PC end, we use `NatNetLinux`[3], a lightweight library built on NatNet API that reads the NatNet UDP packets in Unix-based OSs, to receive the vision data from the Vision PC.

Because the WAM control thread runs at a higher rate (typically 500 Hz) than the vision system (maximum 360 Hz), we developed a two threads structure with a fixed-size RAM buffer to transmit data between them (see in Fig. 3.3). The vision data thread is a non-realtime thread waiting for new UDP packets from specified IP address. As long as a new packet arrives, it will decode the vision data, timestamp it with the current system time and store it to the buffer. As described in Sec. 3.2.1, for the current system there is no way to accurately timestamp capture time to the vision data. The idea of this method is to compensate the delay from data processing and communication by vision data filtering, which will be detailed in the next section. In the WAM control loop the raw vision data is read from the buffer, filtered and estimated object poses. Fig. 3.4 shows the data flow between the primary C++ classes of the program running at the WAM PC.

[3] `NatNetLinux` is written and maintained by Philip G. Lee, more details can be found at `https://github.com/rocketman768/NatNetLinux`.

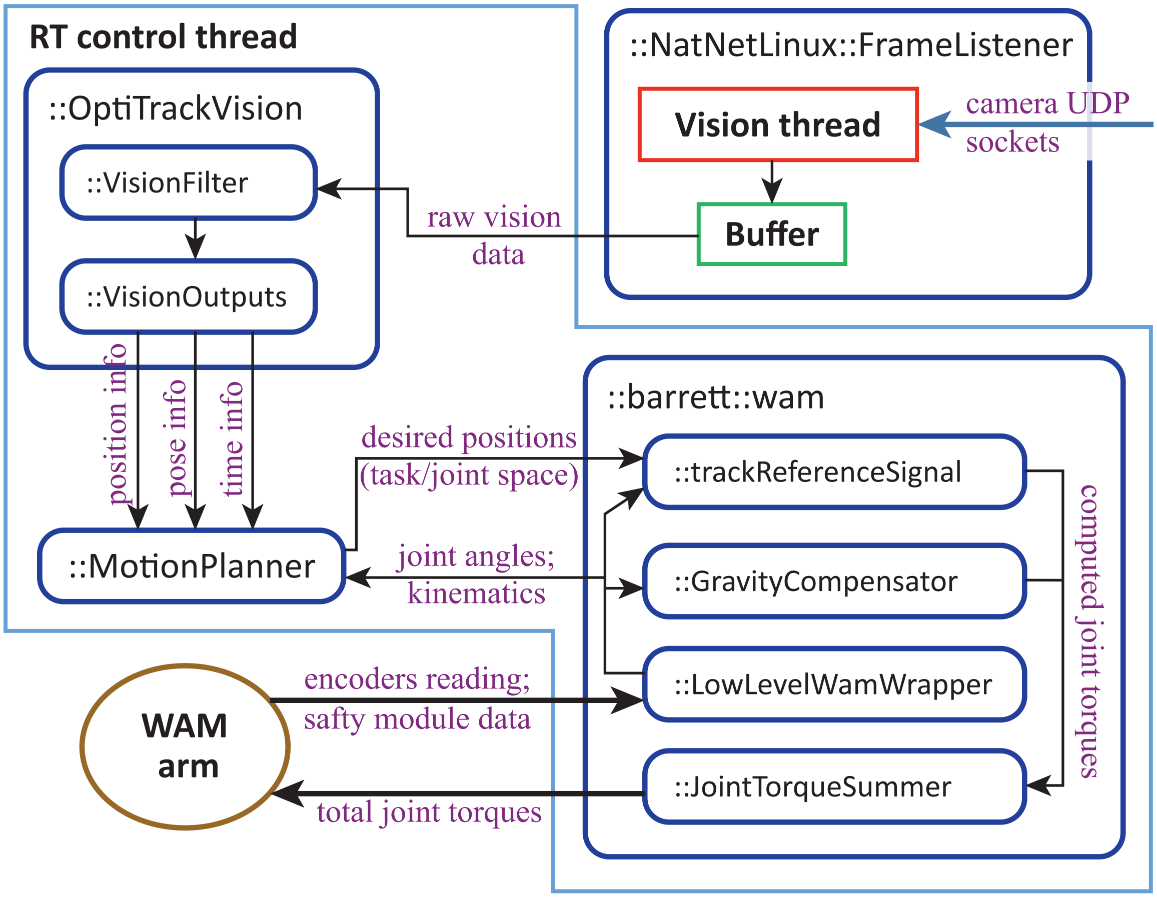
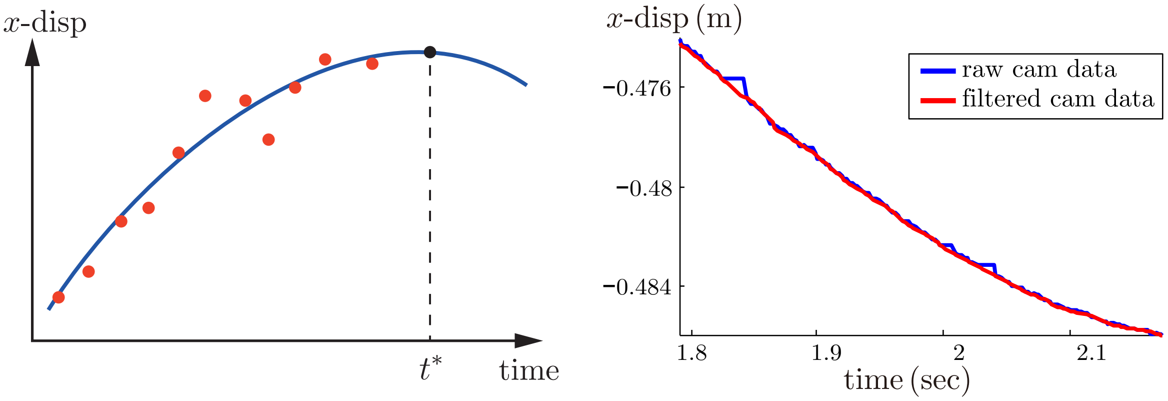


Figure 3.4. The general data flow of the multi-threaded program running on the WAM PC. Rounded rectangles represent C++ classes.

3.3. Vision Data Filtering

In this section we focus on the vision data filtering problem. In general, the design of the filter need to satisfy the following three criteria:

(1) **Incorporating time delay** – As mentioned in Sections 3.2.1 and 3.2.2, since there is no way to accurately timestamp the capturing time with the current system, the time delay caused by Ethernet communication and vision data processing need to be incorporated.



(a) Quadratic curve fitting.

(b) Experiment result.

Figure 3.5. Vision filter diagram and experiment result shown in x-direction.

(2) **Feed data to the realtime thread** – Solving the problem of passing vision data from a non-realtime thread to a realtime thread with a different operating frequency. As mentioned in Section 3.2.2, the vision data thread is a non-realtime thread and running slower than the WAM control thread.

(3) **Realtime calculation** – The goal of filter is to supply object poses as a real time feedback to the control loop, so another criteria is that the algorithm must be computationally efficient — i.e., the computation time of each cycle should be at least smaller than 2 ms as the typical control loop runs at 500 Hz. we found that in each control loop, main tasks including reading joint angles and robot status, calculating desired joint torque and logging data usually take about 0.6 ms $\sim$ 1.2 ms depending on the complexity of the motion planning algorithm. Therefore a reasonable goal of the filter calculation time is around 0.5 ms or less.

Based on the design criteria above, the proposed filter algorithm and its implementation result are shown in following subsections.

3.3.1. Filter Algorithm

The basic idea of the filter algorithm is to fit n most recent history data to a quadratic curve with respect to time t, and then calculate the current position value with current time. The number n need to be greater or equal to three since a quadratic curve need to be uniquely described by at least three parameters. Fig. 3.5a illustrates the idea with the x-direction displacement of an object: the red dots represent the received position data x_i ($i \in 1...n$) at corresponding received time $t_{rec,i}$ on the WAM PC; and at time t^* the control thread requested feedback data and then n of the most recent pairs of the x_i and $t_{rec,i}$ is pulled out from the buffer (sized $m, m \geq n$) to fit into a quadratic curve $x(t)$; and finally plug in the current time t^* to $x(t)$ and calculate the current estimated position $x(t^*)$. In practise all the 6 configurations of the object in 3D space is filtered individually with the same algorithm as describe above. Quadratic is chosen since it is the lowest-order polynomial that can smooth the data therefore minimize the calculation time.

The philosophy behind this algorithm is assuming that during a small time period around the current time, the acceleration of the object remains the same, and the motion of the object can be predicted based on history data. Therefore we call the filter *Constant Acceleration Filter* (CAF).

The CAF can meet the design criteria effectively: the CAF incorporates feedback data with a different rate than the control loop and even when the feedback is delayed for a relative long period, the CAF can still reasonably predict the feedback value from the received data. Meanwhile the CAF is also fast. As we recorded, the average calculation time for CAF with $n = 10$ is around 0.1 ms, which is under the desired calculation time 0.5 ms.

3.3.2. Result

Fig. 3.5b shows the result of the x position tracking of an object that moves randomly in space. The cameras are running at 360 Hz and the filter function was called at 500 Hz in the control loop. For the CAF we used $m = n = 10$. From the result we could tell the raw data is smoothed out after filtering.

3.4. Soft-catch Example

To test the performance of the ERIN system, we implemented a dynamic manipulation task: soft-catching a ball. The task can be described as: a ball is thrown by a user, and the robot arm with a plat palm mounted at the end try to catch and balance the ball to the center of the palm. The word *soft-catch* means that at the moment that the ball touch the palm, both the position and velocity of robot end-point will be the same with the ball's to minimize bouncing. In this example, we substituted the Allgegro hand with a plastic board to reduce the weight in order to achieve higher acceleration of the end-effector.

The problem is challenging since the trajectory of the ball must be estimated in real time . And the motion of the robot arm should also be planned in real time as well.

The center position of the ball in Cartesian space is denoted as $\mathbf{p}_{\text{ball}} = [x_{\text{ball}}, y_{\text{ball}}, z_{\text{ball}}]^T$, and the trajectories of ball $\mathbf{p}_{\text{ball}}(t)$ in free flight are described by quadratic time polynomials assuming the total external forces are constant. we denote z-direction along the gravity direction. At the beginning of the task, whether the ball is in free flight or not is detected by checking whether the z-direction acceleration of the ball $\ddot{z}_{\text{ball}}$ exceed a threshold. And when the ball is in the free flight phase, an algorithm will calculate the trajectory of the ball $\mathbf{p}_{\text{ball}}(t)$ by fitting in its history position data of free flight.

For the motion of the robot arm, we denote the position of the end-point frame as $\mathbf{p}_{\text{arm}} = [x_{\text{arm}}, y_{\text{arm}}, z_{\text{arm}}]^T$. Before the ball is thrown out, the arm remains at the same initial position. When the ball is in free flight, the motion of the end-point position $\mathbf{p}_{\text{arm}}(t)$ are described by polynomials w.r.t. time, among them $x_{\text{arm}}(t)$ and $y_{\text{arm}}(t)$ are defined as cubic polynomials and $z_{\text{arm}}(t)$ is defined as a quadratic polynomial. Then $\mathbf{p}_{\text{arm}}(t)$ can be calculated by solving following equations:

$$\mathbf{p}_{\text{arm}}(t^*) = \mathbf{p}_{\text{arm,current}} \tag{3.1}$$

$$\dot{\mathbf{p}}_{\text{arm}}(t^*) = \dot{\mathbf{p}}_{\text{arm,current}} \tag{3.2}$$

$$\mathbf{p}_{\text{arm}}(t^* + \Delta t) = \mathbf{p}_{\text{ball}}(t^* + \Delta t) \tag{3.3}$$

$$\dot{\mathbf{p}}_{\text{arm}}(t^* + \Delta t) = \dot{\mathbf{p}}_{\text{ball}}(t^* + \Delta t) \tag{3.4}$$

where t^* means the time when the calculation is executed and Δt is the time interval from t^* to the contact time. Eq. (3.1) and (3.2) shows the initial conditions of the end-point trajectories, and Eq. (3.3) and (3.4) shows the final conditions at the contact time, both the position and velocity of the arm end-point should match with the ball's. From Eq. (3.1)–(3.4) we have twelve constraints and by the definition of $\mathbf{p}_{\text{arm}}(t)$ and Δt we have total $4 + 4 + 3 + 1 = 12$ unknowns. Therefore the motion of the arm can be uniquely solved at each control loop.

After the contact is formed, the robot arm will decelerate the ball to rest and start palming. The word *palming* means to balance the ball in the palm and prevent it from falling off. The deceleration motion of the robot arm is given as to stop the ball in a pre-defined distance d_{stop} from the point when the contact was formed. Then the reference acceleration of the robot arm end-point can be calculated as

$$\ddot{\mathbf{q}}_{\text{arm}} = 1/2d_{\text{stop}}[\dot{x}_{\text{arm}}(t_c)^2, \dot{y}_{\text{arm}}(t_c)^2, \dot{z}_{\text{arm}}(t_c)^2]^T, \tag{3.5}$$

where t_c refers to the contact time. And at the same time, the palming algorithm starts to calculate the desired orientation of the palm to let the ball stay in the palm. The palming algorithm is a PD controller uses x, y position errors to calculate the desired orientation of the palm. The reference ball position is specified at the center of the palm so the position error is calculated by subtracting the current end-point position with the ball position.

We implemented the algorithms mentioned above and the result of a successful run can be found in Fig. 3.6. The ball we used is a pressure ball with a diameter of 2.5 in, and the palm we used is a 1×1 ft plastic pad. The task is challenging in terms of the workspace limit and motor torque and speed limits of the WAM arm. Soft-catching will fail in mostly following cases: calculated catching/resting points go out of the workspace, and calculated arm trajectories $\mathbf{q}_{\text{arm}}(t)$ exceed joint velocity/acceleration limits.

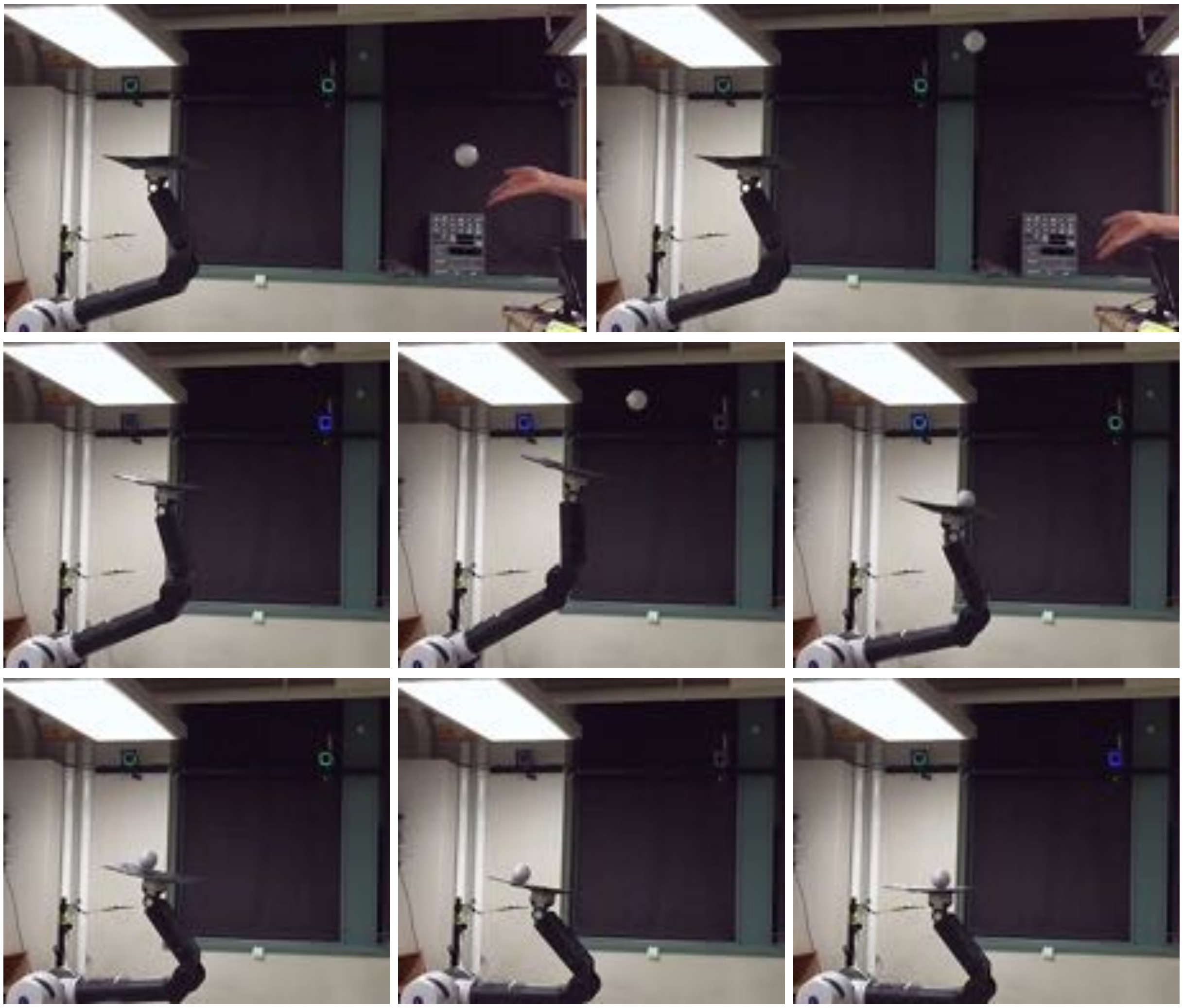


Figure 3.6. Experiment result of soft-catch and palming a ball.

CHAPTER 4

Dynamic In-hand Sliding Manipulation

In this chapter we investigate the problem of regrasping an object by inertia loads. Our testbed is the ERIN robot manipulation system as shown in Chapter 3. Assuming the fingers are compliantly mounted, and the initial grasp configuration is chosen, current research problems include:

(1) given the state of the hand and object, the contact normal forces, and the acceleration of the hand, find the relative acceleration of the object (forward dynamics);

(2) given the state of the hand and object and the desired relative acceleration of the object, find appropriate hand accelerations and contact normal forces (inverse dynamics);

(3) plan the hand motion (and possibly contact normal forces) to achieve a desired regrasp;

(4) repeatedly plan and execute hand motions to iteratively reduce grasp error;

(5) use real-time feedback control of hand motion and finger normal forces during sliding motion to achieve the desired regrasp; and

(6) estimate friction properties from observed hand and object motions, given the contact normal forces.

In this study, we use a simple, spring-actuated passive hand in place of the Allegro hand and tactile sensors, and we study items 1)–4) above. In particular, we focus on the case of planar motion, where the laminar object moves with three degrees of freedom (two translational and one rotational) and the fingers contact the object on opposite sides that are parallel to the plane of motion. Friction property estimation and feedback control exceeds the scope of this paper and will be addressed in future work.

The rest of this chapter can be summarized as following: In Section 4.1 we solve problems 1)–4) for a simple 1-DOF example, as a template for the more general case. In Section 4.2 we generalize the problem statement to an n-fingered grasp moving in a plane. In Section 4.3 we discuss the limit surface model for friction and derive expressions for the frictional wrench given the sliding velocity of an object in an n-fingered grasp consisting of patch contacts. In Section 4.4 we derive the sliding dynamics within the motion plane and outline a method to calculate the acceleration of the object relative to each finger given the accelerations of the fingers. In Section 4.5 we solve the finger motion planning problem for a given n-fingered grasp to achieve a desired regrasp.

The material in Sections 4.2–4.5 solves the planar regrasp problem for general n-fingered grasps of an object with parallel faces. The details of the finger normal forces and grasp limit surface depend on the particular grasp configuration, however. In Section 4.6 we derive the details of the grasp limit surface for a particular type of grasp, a two-fingered pinch grasp. In Section 4.7 we implement the motion planning algorithm for in-hand manipulation with a two-fingered pinch grasp, and we show experimentally that iterative planning and execution can further reduce error in the final grasp configuration.

4.1. 1-DoF Example

In this section we address research topics 1)–4) from the introduction for a 1-DOF example with no gravity. This example serves as a template for the more general problem beginning in Section 4.2.

Consider an object that accelerates in the positive or negative direction due to frictional contact with a single finger. Based on a Coulomb friction coefficient μ and a normal force f_N, the finger can provide a tangential force to the object of up to μf_N before sliding.

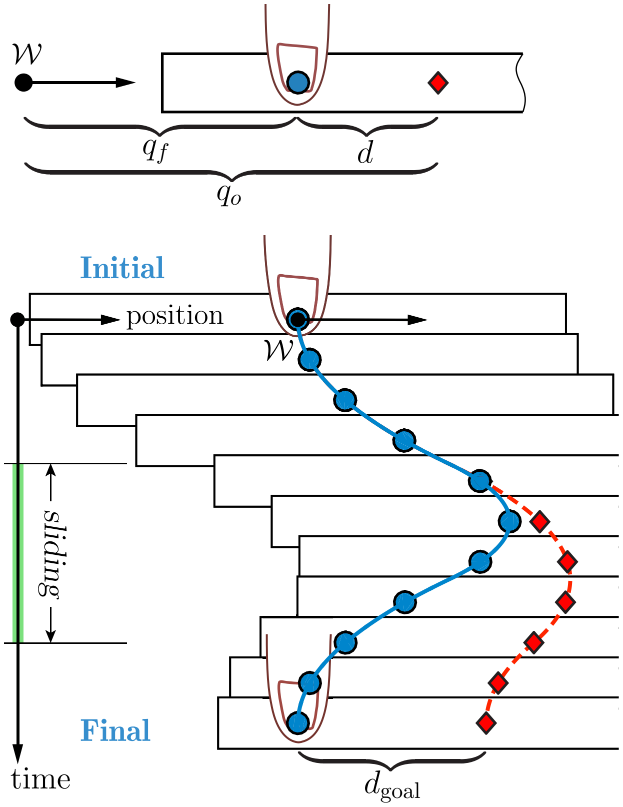


Figure 4.1. (Top) Configuration of the 1-DOF system. The red diamond shows the center of mass (CM) of the object and the blue dot shows the contact point of the finger. (Bottom) An example of in-hand sliding of the 1-DOF system with initial condition $q_f(0) = q_o(0) = 0$. The finger initially accelerates to the right, and then accelerates to the left causing the finger to slide on the object and achieve a desired position relative to the object CM d_{goal}. The corresponding acceleration, velocity, and position profiles are shown in Figure 4.2.

We assume the object has unit mass, so the maximum object acceleration is $a_o = \mu f_N$. We also assume the finger is capable of a maximum acceleration $a_f > a_o$. Additionally we define a finger acceleration a greater than 0 but less than a_o. The relationship between the accelerations can be written as $a_f > a_o > a > 0$.

Let $q_f(0) = q_o(0)$ be the initial position of the finger and the object w.r.t. the world frame $\mathcal{W}$ respectively, and let $d(t) = q_o(t) - q_f(t)$ be the object position relative to the finger position at time t. The problem is to choose a finger acceleration profile $\ddot{q}_f : [0,T] \rightarrow \mathbb{R}$ that causes the object to slide relative to the finger by d_{goal} at time T, i.e., $d(T) = q_o(T) - q_f(T) = d_{\text{goal}}$ as shown in Figure 4.1. Without loss of generality, assume $d_{\text{goal}} > 0$. Similar reasoning applies for the case $d_{\text{goal}} < 0$.

4.1.1. Forward Dynamics

The forward dynamics problem is to determine the relative sliding acceleration $\ddot{d}$ given a finger acceleration $\ddot{q}_f$. If $\dot{d} \neq 0$, then $\ddot{d} = \text{sgn}(\dot{d})a_o - \ddot{q}_f$. If $\dot{d} = 0$ and $|\ddot{q}_f| \leq a_0$, no sliding occurs ($\ddot{d} = 0$). If $\dot{d} = 0$ and $|\ddot{q}_f| > a_0$, then $\ddot{d} = \text{sgn}(\ddot{q}_f)a_0 - \ddot{q}_f$.

4.1.2. Inverse Dynamics

The inverse problem is to determine the finger acceleration $\ddot{q}_f$ that achieves a desired relative sliding acceleration $\ddot{d}$. If $\dot{d} \neq 0$, then $\ddot{q}_f = \text{sgn}(\dot{d})a_o - \ddot{d}$. If $\dot{d} = \ddot{d} = 0$, no slip occurs so any $|\ddot{q}_f| \leq a_o$ is valid. If $\dot{d} = 0$ and $|\ddot{d}| > 0$ (you are trying to initiate slip), then $\ddot{q}_f = \text{sgn}(\ddot{d})a_0 - \ddot{d}$.

4.1.3. Motion Planning

We assume the finger and object are initially at rest and $q_f(0) = q_o(0) = 0$, and require that the finger's net displacement and velocity after the motion are zero. To achieve the sliding regrasp while satisfying these constraints, first accelerate the finger with $\ddot{q}_f = a$ for time T_1. Then apply the maximum negative acceleration $\ddot{q}_f = -a_f$ for time T_2. Next apply $\ddot{q}_f = a$ for time $T_3 + T_4 = T_{34}$. To achieve zero final displacement and velocity for the finger, we choose $T_1 = T_{34}$ and $\dot{q}_f(T_1) = -\dot{q}_f(T_1 + T_2)$.

The motion plan consists of three phases: an initial sticking phase of duration T_1, a sliding phase of duration $T_2 + T_3$, and a final sticking phase of duration T_4. During phase

1 $(0 \leq t < T_1)$, $\dot{d}$ is zero and $|\ddot{q}_f| \leq a_o$ so no relative motion occurs. During the first part of phase 2 $(T_1 \leq t < T_1 + T_2)$, the negative acceleration is sufficiently high that sliding occurs $(\ddot{q}_f < -a_o)$. During the second part of phase 2 $(T_1 + T_2 \leq t < T_1 + T_2 + T_3)$, the acceleration magnitude is decreased $(|\ddot{q}_f| \leq a_o)$ but $\dot{d} \neq 0$ so sliding still occurs until $\dot{d} \to 0$. During phase 3 $(T_1 + T_2 + T_3 \leq t < T_1 + T_2 + T_3 + T_4)$, the object is sticking and $\dot{d} = 0$. Figure 4.1 shows an example of in-hand sliding of the 1-DOF system. The full series of accelerations, resulting velocities, and positions are shown in Figure 4.2.

The total relative sliding distance d_{goal} is the integral between the finger and object velocity curves in the sliding phase. With given values of a_f, a_o, a and d_{goal}, we solve the following constraints to find the durations T_1, T_2, T_3, and T_4:

$$2aT_1 = a_f T_2, \tag{4.1}$$

$$a_o(T_2 + T_3) = a(2T_1 - T_3), \tag{4.2}$$

$$d_{\text{goal}} = 0.5(a_f - a_o)(T_2^2 + T_2T_3), \tag{4.3}$$

$$T_4 = T_1 - T_3. \tag{4.4}$$

Equation (4.1) enforces that the finger velocity at time $T_1 + T_2$ is the opposite of the finger velocity at time T_1. Equation (4.2) requires the object to stop slipping relative to the finger at time $T_1 + T_2 + T_3$. Equation (4.3) enforces the desired slipping distance of the object. Together, Equations (4.1), (4.2) and (4.4) ensure the total finger displacement is zero after the regrasp motion. The chosen constraints ensure that T_1, T_2, T_3, and T_4 can be solved for analytically which simplifies the 1-DOF problem. We can solve Equations (4.1)–(4.3)

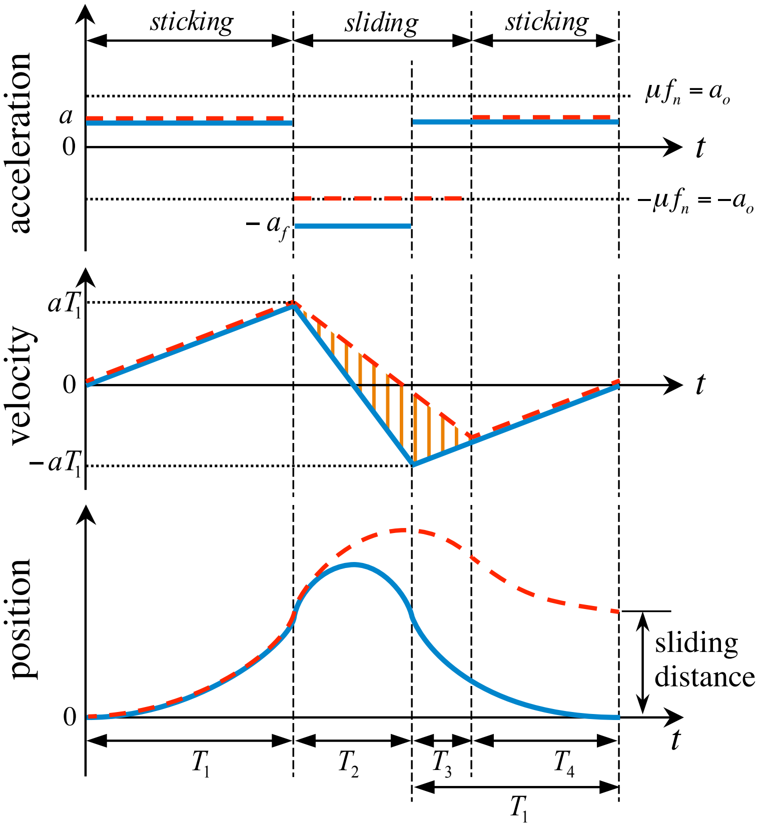


Figure 4.2. A plot of the system motion profile of the 1-DOF problem. The solid blue curves represent the motion of the finger and the dashed red curves represent the motion of the object. The object has unit mass. In the velocity profile, the orange shaded area is the relative sliding distance.

for T_1, T_2, T_3:

$$\begin{aligned} T_1 &= \frac{a_f}{a}\sqrt{\frac{d_{\text{goal}}(a+a_o)}{2(a+a_f)(a_f-a_o)}}, \\ T_2 &= \sqrt{\frac{2d_{\text{goal}}(a+a_o)}{(a+a_f)(a_f-a_o)}}, \\ T_3 &= \sqrt{\frac{2d_{\text{goal}}(a_f-a_o)}{(a+a_f)(a+a_o)}}. \end{aligned} \tag{4.5}$$

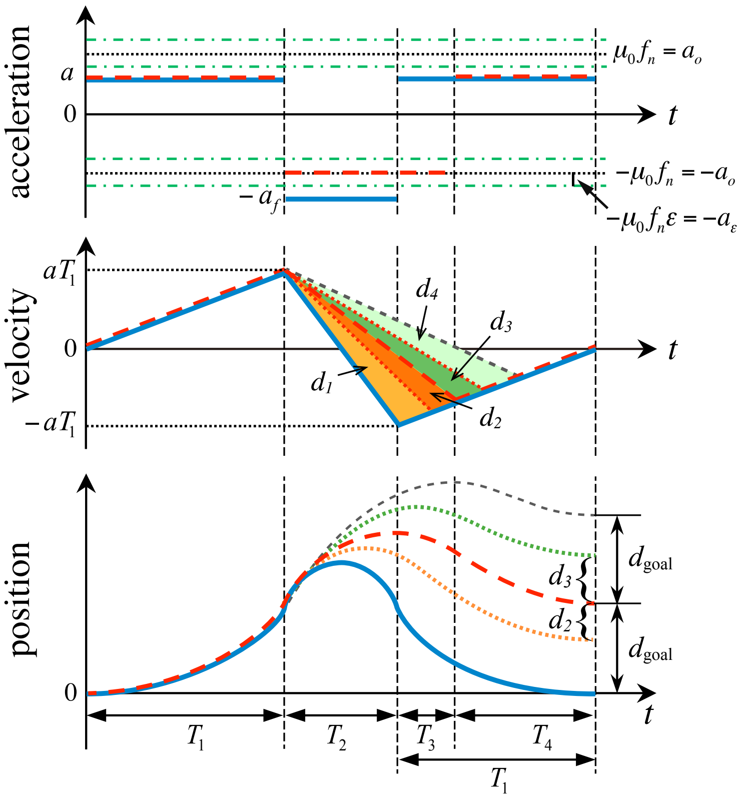


Figure 4.3. Friction uncertainty affects the sliding distance. We denote d_1, d_2, d_3, d_4 as the areas of different triangles and $d_1 + d_2 = d_3 + d_4 = d_{\text{goal}}$. The areas d_2 and d_3 show the uncertainty in the sliding distance. Area d_2 represents the error when the friction coefficient is underestimated, and d_3 represents the overestimated case.

4.1.4. Iterative Error Reduction

Following the execution of a planned repositioning trajectory, there will be some error in the actual relative displacement due to trajectory tracking error, errors in initial conditions, or unmodeled dynamics. A significant source of error is an incorrect estimate of the friction coefficient μ. The following theorem shows that iterated executions of motion plans based on updated displacement information are sufficient to bring the object to the desired goal position d_{goal} in the presence of significant uncertainty in the friction coefficient.

Theorem 1. *Consider the 1-DOF sliding regrasp system with a desired net sliding distance d_{goal}, a known constant normal force f_N, an estimated friction coefficient μ_0, and an actual (unknown) constant friction coefficient $\mu \in [\mu_0(1-\epsilon), \mu_0(1+\epsilon)]$ for a friction coefficient uncertainty $0 < \epsilon < 2/3$.*

For any acceleration a in the range $\mu_0 f_N(2\epsilon - 1) < a < \mu_0 f_N(1-\epsilon)$, and for any positive constant ρ satisfying $\frac{\mu_0 f_N \epsilon}{a+\mu_0 f_N(1-\epsilon)} < \rho < 1$, by iterating the finger motion described in Section 4.1.3 (where d_{goal} is recalculated at each iteration based on perfect sensor data), the error in the net sliding distance converges exponentially to zero at least as fast as ρ^k converges to zero as the iteration number k goes to infinity, provided

$$a_f \geq \frac{\mu_0 f_N[\mu_0 f_N(1-\epsilon) + a(\epsilon/\rho + 1)]}{a - \mu_0 f_N[\epsilon(1+1/\rho) - 1]}.$$

Proof: See Appendix A.

Remark 1. *From the condition on a_f given in Theorem 1, as the chosen value of ρ gets smaller, the required minimum value of a_f increases to ensure the convergence property.*

The following example shows how choices of a, ρ, and a_f affect the iterative reduction algorithm. For a given $a_o > 0$ and $\epsilon = 0.2$, we choose $a = 0.4a_o$. The feasible range of ρ is $0.167 < \rho < 1$ according to the ρ constraints from Theorem 1. Note that the value of ρ determines the basic error convergence rate, and also affects the lower bound of the maximum finger acceleration a_f.

In practice the choice of ρ and a_f should be based on manipulator acceleration constraints. Choosing $\rho = 0.5$ requires a finger acceleration a_f to be at least $1.7a_0$ to ensure that the error can be driven to zero at the rate ρ^k. Figure 4.4 illustrates how the net

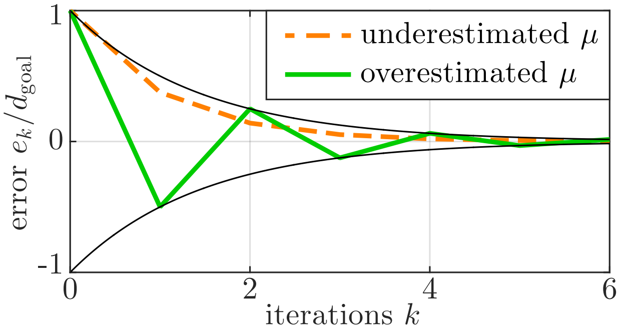


Figure 4.4. Iterative reduction of the net error in sliding distance. The worst-case scenarios are shown, where the actual friction coefficient is at the extreme $\mu = \mu_0(1 - \epsilon)$ and $\mu_0(1 + \epsilon)$. The solid black curves show the bounding convergence rate $\pm\rho^k$.

sliding distance error converges to zero by iterating the motion planning in worst case scenarios $\mu = \mu_0(1 - \epsilon)$ and $\mu = \mu_0(1 + \epsilon)$.

4.2. General Problem Statement

In this section we generalize the in-hand manipulation problem outlined in Section 4.1 to an n-fingered grasp, and define notation used in the rest of the paper.

We assume the object to be a laminar part that moves in a plane, held by n patch-contact fingers located on opposite sides of the part. The laminar part, and all motion of the part, are in a plane fixed at an angle α relative to a horizontal plane orthogonal to the gravity direction. A fixed frame $\mathcal{W}$ is defined in the plane of motion such that its x and y axes are basis vectors for the plane of motion and the y-axis is opposite to the projection of the gravity vector to the motion plane $\mathbf{g}_{\|} = [0, -mg\sin\alpha]^{\mathsf{T}}$, as shown in Figure 4.5.

The mass of the object is denoted m, and its scalar inertia about its CM is I. The sum of out-of-plane forces applied to the object satisfies force balance at all instances so the object remains in the xy-plane of frame $\mathcal{W}$. We assume we can control the acceleration of each finger.

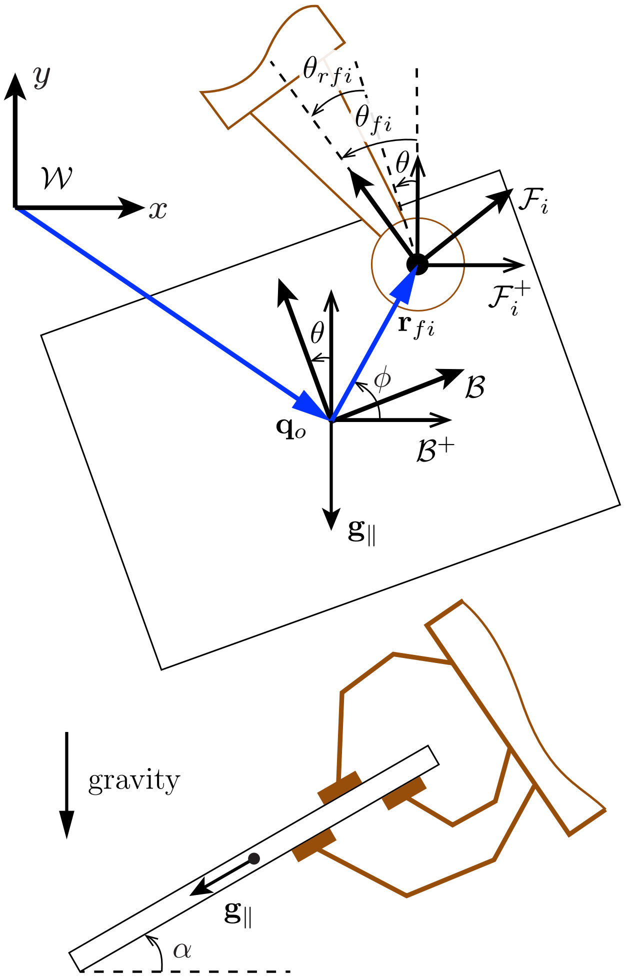


Figure 4.5. Laminar object grasped by n patch contact fingers. (Top) View of the system in the object plane. The variables are defined in Section 4.2. (Bottom) The angle between the object plane and the horizontal plane is denoted as α.

Frame $\mathcal{B}$ is the body frame fixed to the CM of the object, and the $x^{\mathcal{B}}$ and $y^{\mathcal{B}}$ axes are in the plane of the object. The finger contact patches are assumed circular. The frame $\mathcal{F}_i$ of the ith finger is located at the center of the finger's contact patch. We denote $\mathcal{B}^+$ and $\mathcal{F}_i^+$ as frames where the origins are coincident with $\mathcal{B}$ and $\mathcal{F}_i$ respectively and the axes are aligned with $\mathcal{W}$. All configurations and velocities are defined with respect to the world frame $\mathcal{W}$ unless noted otherwise. All vectors are written in bold lowercase letters, all matrices are written in bold capital letters, and scalars are written in italic letters. We denote the configuration of the object by its pose $\mathbf{q}_o = [x, y, \theta]^\mathsf{T}$, representing the position and orientation of $\mathcal{B}$ relative to $\mathcal{W}$. The location of the frame $\mathcal{F}_i$ is $\mathbf{q}_{fi} = [x_{fi}, y_{fi}, \theta_{fi}]^\mathsf{T}$, and the entire n-fingered grasp is defined as $\mathbf{q}_f = [\mathbf{q}_{f1}^\mathsf{T}, \ldots, \mathbf{q}_{fn}^\mathsf{T}]^\mathsf{T}$. The relative positions between the object and the finger contacts are defined as $\mathbf{r}_{fi} = [x_{rfi}, y_{rfi}, \theta_{rfi}]^\mathsf{T}$, where $\mathbf{q}_{fi} = \mathbf{q}_o + \mathbf{r}_{fi}$, and the relative position for the entire grasp is defined as $\mathbf{r}_f = [\mathbf{r}_{f1}^\mathsf{T}, \ldots, \mathbf{r}_{fn}^\mathsf{T}]^\mathsf{T}$. The configuration and velocity of the system are denoted as $\mathbf{q} = [\mathbf{q}_o^\mathsf{T}, \mathbf{q}_f^\mathsf{T}]^\mathsf{T}$ and $\dot{\mathbf{q}} = [\dot{\mathbf{q}}_o^\mathsf{T}, \dot{\mathbf{q}}_f^\mathsf{T}]^\mathsf{T}$. The full state of the system is defined as $[\mathbf{q}^\mathsf{T}, \dot{\mathbf{q}}^\mathsf{T}]^\mathsf{T}$. Figure 4.5 shows an example of a planar system with a rectangular object and patch contact fingers.

4.3. Frictional Limit Surface

In this section we discuss the concept of frictional limit surfaces (LS) and how they are shaped given circular patch contacts. Additionally we derive expressions for the frictional force applied to the object from a patch contact with a given finger velocity relative to the object, and for the grasp limit surface given n individual limit surfaces.

4.3.1. Patch Contact

In this paper, friction is assumed to conform to Coulomb's law. For a circular patch contact, we denote $\mathbf{f} = [f_x, f_y, m_z]^\mathsf{T}$ as the frictional wrench applied to the object expressed in frame $\mathcal{F}^+$. To describe the boundary of the frictional wrenches given the contact normal force, we use the concept of a frictional limit surface [19, 20, 21]. Frictional limit surfaces are convex and closed. When the frictional wrench $\mathbf{f}$ lies within the LS, the finger sticks on the object; and if the finger slides with velocity $\mathbf{v}$ relative to the object, the frictional wrench $\mathbf{f}_c$ is on the LS at a location where $\mathbf{v}$ is normal to the LS at $\mathbf{f}_c$ to satisfy the maximum work inequality (Figure 4.6).

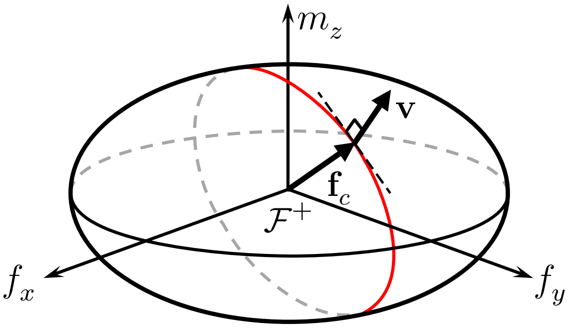


Figure 4.6. Ellipsoid limit surface expressed in a local frame $\mathcal{F}^+$ attached to the center of the contact. The sliding direction $\mathbf{v}$ is along the normal of the ellipsoid at the corresponding frictional wrench $\mathbf{f}_c$.

The LS for a soft-finger contact can be approximated by an ellipsoid in the local frame [**66**]. A mathematical representation of the LS is given by the following quadratic form expressed in a local frame $\mathcal{F}^+$:

$$\mathbf{f}^\mathsf{T}\mathbf{A}\mathbf{f} = 1, \tag{4.6}$$

where the matrix $\mathbf{A} \in \mathbb{R}^{3\times 3}$ is a symmetric positive-definite matrix that determines the shape of the LS ellipsoid.

During sliding, the frictional wrench $\mathbf{f}_c$ lies on the LS, and we can write the relative velocity $\mathbf{v}$ along the direction of the gradient of the ellipsoid with respect to $\mathbf{f}$ at $\mathbf{f}_c$ as

$$\mathbf{v} = \lambda \, \frac{\partial}{\partial \mathbf{f}} \Big(\mathbf{f}^\mathsf{T}\mathbf{A}\mathbf{f} \Big) \Big|_{\mathbf{f}_c} \tag{4.7}$$

for some $\lambda \in \mathbb{R}$ which scales the normal vector to the relative velocity vector. For a given relative velocity, the corresponding frictional wrench can be written as

$$\mathbf{f}_c = \frac{1}{\lambda}\mathbf{B}\mathbf{v}, \tag{4.8}$$

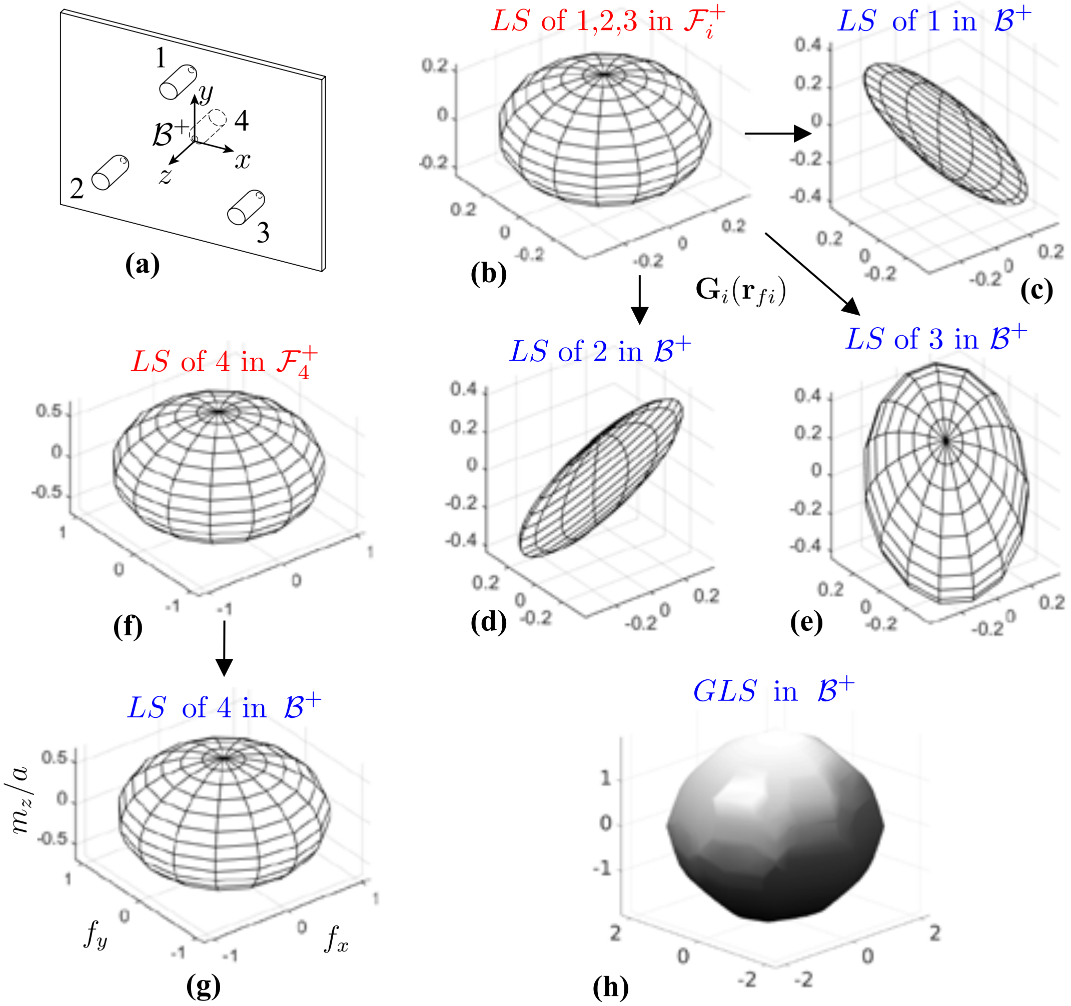


Figure 4.7. A 4-fingered grasp and the resulting limit surfaces in the local finger frames, the body frame, and the composite grasp limit surface. (a) is a sketch of the grasp with three fingers on one side and one on the opposite side. (b) identical limit surface for fingers 1–3 in the local finger frames $\mathcal{F}_i^+$. (c), (d), and (e) limit surfaces from fingers 1, 2, and 3 respectively mapped to the common frame $\mathcal{B}^+$ using the $\mathbf{G}_i$ transformation. (f) limit surface for finger 4, which is the same in $\mathcal{F}_4^+$ as $\mathcal{B}^+$ since they are coincident. (h) composite grasp limit surface. The axes in (b)–(g) are all aligned and equivalent to the axes in (g).

where $\mathbf{B} = \frac{1}{2}\mathbf{A}^{-1}$. Substituting Equation (4.8) into Equation (4.6) and utilizing $(\mathbf{A}^{-1})^\mathsf{T} = \mathbf{A}^{-1}$, we have

$$\lambda = \frac{1}{2}\sqrt{\mathbf{v}^\mathsf{T}\mathbf{A}^{-1}\mathbf{v}}. \tag{4.9}$$

Combining Equations (4.8) and (4.9) we derive the function $\Gamma(\cdot)$ which gives the frictional wrench as a function of a given relative velocity $\mathbf{v}$:

$$\mathbf{f}_c = \Gamma(\mathbf{v}) = \frac{\mathbf{A}^{-1}\mathbf{v}}{\sqrt{\mathbf{v}^{\mathsf{T}}\mathbf{A}^{-1}\mathbf{v}}}. \tag{4.10}$$

4.3.2. Grasp Limit Surface

When multiple fingers contact an object, the individual LS can be mapped to a common frame to generate the grasp limit surface (GLS). Let $\mathbf{f}_i$ represent the frictional wrench applied to the object expressed in the local frame $\mathcal{F}_i^+$. A reasonable choice of a common frame is the frame $\mathcal{B}^+$. The 3×3 matrix $\mathbf{G}(\mathbf{r}_{fi})$ is the map relating the frictional wrench $\mathbf{f}_i$ in $\mathcal{F}_i^+$ to the wrench expressed in $\mathcal{B}^+$. Matrices $\mathbf{G}(\mathbf{r}_{fi})$ depend on the contact position relative to the object CM and is defined as

$$\mathbf{G}(\mathbf{r}_{fi}) = \mathbf{G}_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -y_{rfi} & x_{rfi} & 1 \end{bmatrix}. \tag{4.11}$$

The grasp limit surface is the convex hull of the sum of all possible friction forces that the grasp can resist. The GLS can be expressed in $\mathcal{B}^+$ as

$$GLS = \delta\{\mathbf{f} | \mathbf{f} = \sum_{i=1}^{n} \mathbf{G}_i \mathbf{f}_i \ \forall \ \mathbf{f}_i \in LS_i\}, \tag{4.12}$$

where δ is an operator that takes the boundary of the set, $\mathbf{f} = [f_x, f_y, m_z]^{\mathsf{T}}$ is an arbitrary friction force on the GLS, and LS_i is the limit surface for contact i.

Figure 4.7 shows an example of a four-fingered grasp on an object and the resulting limit surfaces with frictional torques about the corresponding contact center, transferred frictional torques expressed in $\mathcal{B}^+$, and the combined grasp limit surface.

4.4. Dynamics

In this section we derive the dynamics for the case where the object is sticking and when it is sliding. We assume that the system state $[\mathbf{q}^{\mathsf{T}}, \dot{\mathbf{q}}^{\mathsf{T}}]^{\mathsf{T}}$, the matrices $\mathbf{A}_i$ that determine the shape of LS_i, and either the desired relative finger accelerations $\ddot{\mathbf{r}}_{fi}(t)$ or the finger accelerations $\ddot{\mathbf{q}}_{fi}(t)$ are given.

4.4.1. Sticking Dynamics

The object's dynamics are defined as

$$\mathbf{M}\ddot{\mathbf{q}}_o = \sum_{i=1}^{n} \mathbf{G}_i \mathbf{f}_i + \mathbf{g}, \tag{4.13}$$

where $\mathbf{M} = \mathrm{diag}(m, m, I)$ is the mass matrix of the object and $\mathbf{g} = [0, -mg\sin\alpha, 0]^{\mathsf{T}}$ is the wrench on the object due to gravity in the object plane expressed in frame $\mathcal{B}^{+}$. For the sticking case the frictional force at each contact is contained within the limit surface, i.e., $\mathbf{f}_i^{\mathsf{T}}\mathbf{A}_i\mathbf{f}_i < 1$.

4.4.2. Sliding Dynamics

During sliding relative velocity at each contact is defined as

$$\mathbf{v}_i = \dot{\mathbf{q}}_{fi} - \mathbf{G}_i^{\mathsf{T}}\dot{\mathbf{q}}_o. \tag{4.14}$$

The forward dynamics problem is to determine the relative acceleration of each finger $\ddot{\mathbf{r}}_{fi}$ when given the state of the system $[\mathbf{q}, \dot{\mathbf{q}}]^{\mathsf{T}}$, the LS_i shape matrices $\mathbf{A}_i$, and the accelerations of each finger $\ddot{\mathbf{q}}_{fi}$. First we define the relative acceleration as

$$\ddot{\mathbf{r}}_{fi} = \ddot{\mathbf{q}}_{fi} - \ddot{\mathbf{q}}_o. \tag{4.15}$$

The dynamics in Equation (4.13) can be rewritten as

$$\ddot{\mathbf{q}}_o = \mathbf{M}^{-1}\left[\sum_{i=1}^{n}\mathbf{G}_i\mathbf{f}_i + \mathbf{g}\right]. \tag{4.16}$$

Combining Equations (4.10), (4.14), (4.15), and (4.16), we can write the relative finger acceleration as

$$\ddot{\mathbf{r}}_{fi} = \ddot{\mathbf{q}}_{fi} - \mathbf{M}^{-1}\left[\sum_{i=1}^{n}\mathbf{G}_i\Gamma(\dot{\mathbf{q}}_{fi} - \mathbf{G}_i^{\mathsf{T}}\dot{\mathbf{q}}_o) + \mathbf{g}\right]. \tag{4.17}$$

This equation allows us to calculate the relative sliding motion for given finger accelerations, and solves the forward dynamics problem.

The inverse problem is trivial, and Equation (4.17) can easily be rearranged to solve for the required finger accelerations when given a desired relative sliding motion. For the inverse problem it is more convenient to give the relative acceleration w.r.t. the body frame as the input. Denoting $\ddot{\mathbf{r}}_{fi}^{\mathcal{B}}$ as the relative acceleration w.r.t. the body frame $\mathcal{B}$, we have

$$\mathbf{r}_{fi} = \mathbf{T}_i(\theta)\mathbf{r}_{fi}^{\mathcal{B}}, \tag{4.18}$$

where $\mathbf{T}_i(\theta) \in SE(2)$ is the homogeneous transformation which maps $\mathbf{r}_{fi}^{\mathcal{B}}$ into $\mathbf{r}_{fi}$,

$$\mathbf{T}_i(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Taking the first and second derivative with respect to time on both sides of Equation (4.18) gives

$$\dot{\mathbf{r}}_{fi} = \dot{\mathbf{T}}_i \mathbf{r}^{\mathcal{B}}_{fi} + \mathbf{T}_i \dot{\mathbf{r}}^{\mathcal{B}}_{fi}, \tag{4.19}$$

$$\ddot{\mathbf{r}}_{fi} = \ddot{\mathbf{T}}_i \mathbf{r}^{\mathcal{B}}_{fi} + 2\dot{\mathbf{T}}_i \dot{\mathbf{r}}^{\mathcal{B}}_{fi} + \mathbf{T}_i \ddot{\mathbf{r}}^{\mathcal{B}}_{fi}. \tag{4.20}$$

4.5. Motion Planning

In this section, we focus on motion planning to achieve a desired sliding regrasp. For simplicity, we assume that each finger of the hand remains stationary relative to the palm of the hand, so we only plan the motion of the three degrees of freedom of the palm, not individual motions of the fingers.

The motion planning problem can be stated as: given an initial grasp of the object and a desired relative configuration between the object and the hand, find a motion of the hand that achieves this reconfiguration by dynamic in-hand sliding.

The details of the grasp limit surface, and therefore the sliding dynamics, are a function of the number of fingers, their placement on the object, and the normal force control strategy. A specific type of grasp, a two-fingered pinch grasp, is examined in Section 4.6 and used in our experiments.

4.5.1. Specifications for the Motion Planner

- The grasp limit surface details for the specific grasp are given.
- The hand motion yields a sticking phase, followed by a sliding phase, followed by a sticking phase. The time periods for each phase are denoted T_1, T_2, and T_3 respectively, and the total time is denoted $T_{\text{total}} = T_1 + T_2 + T_3$.

- From the given initial grasp and desired relative configuration, the initial and goal relative positions between the object and the fingers $\mathbf{r}^{\mathcal{B}}_{f,\text{init}}$ and $\mathbf{r}^{\mathcal{B}}_{f,\text{goal}}$ can be calculated and are inputs to the motion planner.
- The relative finger trajectory $\mathbf{r}^{\mathcal{B}}_{f}(t)$ in the sliding phase and object trajectories $\mathbf{q}_o(t)$ in the sticking phases are defined as cubic polynomials of time, where each motion component is of the form $a_0+a_1t+a_2t^2+a_3t^3$, defined by four coefficients. Thus the start and end position and velocity provide four constraints on the four coefficients, uniquely defining the polynomial as a function of time.
- The system starts and ends at rest with no relative velocity between the part and object, so $\dot{\mathbf{q}}_o(0) = 0,\ \dot{\mathbf{q}}_o(T_{\text{total}}) = 0,\ \dot{\mathbf{r}}^{\mathcal{B}}_{f}(T_1) = 0,\ \dot{\mathbf{r}}^{\mathcal{B}}_{f}(T_1 + T_2) = 0$.

4.5.2. Planning Algorithm

With the specifications above, the hand and object motion is determined by a set of design variables. The system motion is split into three phases: sticking, sliding, and sticking. In the first sticking phase, we have to choose the object start configuration (initial velocity is zero), end configuration, end velocity, and duration ($3+3+3+1 = 10$ design variables). For the sliding phase, we have to choose only the duration of sliding (one design variable). For the second sticking phase, we have to specify the final object configuration and the duration of the phase ($3 + 1 = 4$ design variables). Thus there is a total of 15 design variables defining a motion plan. As described below, the motion planning problem is turned into a nonlinear root-finding problem to find these 15 variables.

We denote t as the time variable for the entire motion, $t \in [0, T_{\text{total}}]$, and t_i as the time variable for each phase starting at zero and ending at the duration of that phase, $t_i \in [0, T_i],\ i = 1, 2, 3$. The details of the design variables and constraints on each phase are given below.

First sticking phase. The design variables are $\mathbf{q}_o(0), \mathbf{q}_o(T_1), \dot{\mathbf{q}}_o(T_1)$ and T_1. The use of cubic polynomials means there are no freedoms in the trajectory shapes once the boundary conditions are set. Therefore $\mathbf{q}_o(t_1)$ is determined with a given set of the 10 design variables. Note that because the initial relative position is not relevant to where the object is in the world frame, we can choose any initial position $\mathbf{q}_o(0)$. The frictional wrench $\mathbf{f}_i(t_1)$ can be calculated by the sticking dynamics discussed in Section 4.4.1. Since there is no relative motion in the sticking phase, the finger motions $\mathbf{q}_{fi}(t_1)$ are determined as long as $\mathbf{q}_o(t_1)$ and the relative positions $\mathbf{r}^{\mathcal{B}}_{f,\text{init}}$ are given.

The constraints that have to be satisfied are manipulator constraints (including workspace, velocity and acceleration limits) and that the frictional wrenches are always inside the limit surfaces during the first sticking phase.

Sliding phase. The design variable is T_2. In the sliding phase, the cubic polynomial defining the object motion relative to the hand is fully specified by $\mathbf{r}^{\mathcal{B}}_{f,\text{init}}$ and $\mathbf{r}^{\mathcal{B}}_{f,\text{goal}}$ which are given. The initial state of the hand is given by the design variables $\mathbf{q}_o(T_1)$ and $\dot{\mathbf{q}}_o(T_1)$ from the first sticking phase above. To find the hand motion during this sliding phase, we first solve the inverse dynamics using the hand state at the beginning of the trajectory, as well as $\ddot{\mathbf{r}}^{\mathcal{B}}_f$, to find the hand acceleration $\ddot{\mathbf{q}}_f$. Taking a small integration step, we get the next state of the hand, solve the inverse dynamics again, etc., until we have numerically constructed the trajectory of the hand during the sliding phase based on the initial state of the hand and the pre-specified relative object motion during sliding.

The constraints that have to be satisfied in the sliding phase are manipulator constraints.

Second sticking phase. The design variables are $\mathbf{q}_o(T_{\text{total}})$ and T_3. Similar to the first sticking phase, the hand motion $\mathbf{q}_{fi}(t_3)$ and object motion $\mathbf{q}_o(t_3)$ are determined by the specified final conditions and the initial conditions $\mathbf{q}(T_1 + T_2)$ and $\dot{\mathbf{q}}(T_1 + T_2)$ from the final state of the sliding phase.

The constraints that have to be satisfied are manipulator constraints and that the frictional wrenches are always inside the limit surfaces during the second sticking phase.

Full motion planning problem statement.

$$\begin{aligned}
\textbf{given } & \mathbf{r}^{\mathcal{B}}_{f,\text{init}}, \mathbf{r}^{\mathcal{B}}_{f,\text{goal}} \\
\textbf{find } & T_1, T_2, T_3,\ \mathbf{q}_o(0),\ \mathbf{q}_o(T_{\text{total}}),\ \mathbf{q}_o(T_1),\ \dot{\mathbf{q}}_o(T_1) \\
\textbf{such that } & \mathbf{r}^{\mathcal{B}}_{f}(T_1) - \mathbf{r}^{\mathcal{B}}_{f,\text{init}} = 0, \\
& \mathbf{r}^{\mathcal{B}}_{f}(T_1 + T_2) - \mathbf{r}^{\mathcal{B}}_{f,\text{goal}} = 0, \\
& \text{and the dynamics of each phase} \\
& \text{and manipulator constraints are satisfied.}
\end{aligned}$$

This is a multidimensional root-finding problem with constraints, and we use MATLAB's `fmincon` SQP solver to solve it. An initial guess of the design variables is automatically generated based on heuristics encoding our knowledge of the task.

4.6. Limit Surface for a Pinch Grasp

The modeling and motion planning of the previous sections requires the individual fingertip limit surfaces as a function of the configuration of the object relative to the hand, and these limit surfaces depend on the specifics of the grasp. In this paper, we focus on a two-fingered pinch grasp with fingertip patch contacts. To focus on the essential ideas of this paper, the mechanics of sliding regrasp, and iterative motion planning and execution, we built a custom two-fingered passive gripper, shown in Figure 4.8. We use this gripper instead of the Allegro hand for two reasons: (1) it is lighter than the Allegro hand (0.25 kg vs. 1.2 kg), which allows larger accelerations, and (2) it creates a

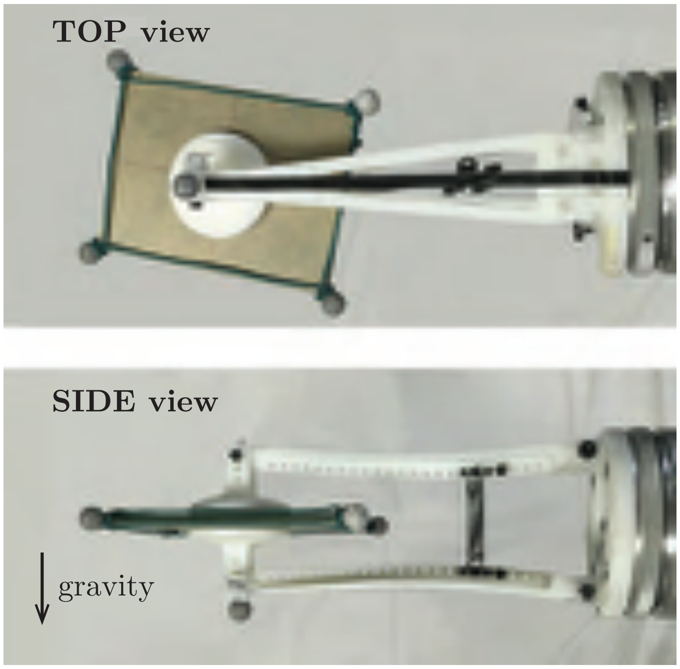


Figure 4.8. The lightweight, spring-powered, constant-gripping-force gripper.

well-characterized constant normal force at the fingertips, allowing us to avoid the potential confounding issue of errors in fingertip force control while we validate the general approach.

In this section we describe the contact model of the constant-gripping-force gripper, and in Section 4.7 we describe regrasp experiments using it.

4.6.1. Pinch-Grasp Description

For this analysis we focus on the case of a zero-thickness planar object pinched by two fingers on opposite sides of the part (Figure 4.9). The two fingers stay stationary relative to each other, and we assume circular contact patches with the same fixed radius a. The plane in which the object and the manipulator move is tilted by angle α from the horizontal plane as shown in Figure 4.5. We denote $f_g = -mg\cos\alpha$ as the gravity force acting on the object in the out-of-plane direction.

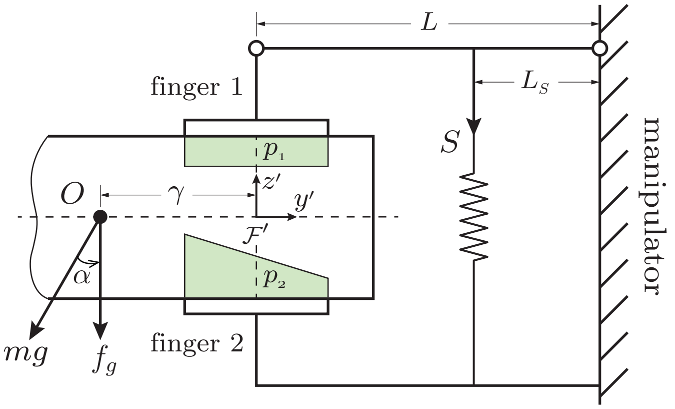


Figure 4.9. Side view of the system. The green shaded regions show the pressure distributions of the contacts.

Finger 1, on the top of the object, is connected to the manipulator by two hinges and an arm of length L. Finger 2, on the bottom of the object, is connected to the manipulator through an arm fixed to the manipulator with the same length L. The two hinges keep the two flat circular fingers parallel to each other and in full contact with the object. The distance from the hinges to the contact of finger 1 is assumed to be zero. The spring is located L_S away from the manipulator and the spring force is denoted S. This model allows us to control the normal forces at the fingers by the spring stiffness (another model could be force control of the manipulator in the normal direction and motion control in the two linear tangent directions and rotation about the contact normal).

4.6.2. Fingertip Limit Surfaces

We first note four important features of the two-fingered pinch grasp:

(1) **Collocated point fingers cannot hold the object.** Remembering that the laminar object is modeled in the limit of zero thickness, the contact point of each finger would be at the same point. Therefore contact forces from the two collocated fingertips always make zero moment about the contact point, and they

cannot balance the moment due to gravity. This issue can be addressed by having contact patches instead of point contacts.

(2) **The pressure distribution at a contact patch is generally unknowable.** If the object and fingertip are modeled as rigid bodies, then the pressure as a function of the location on a continuous contact patch will be indeterminate. Our approach is to use the simplest possible model of the pressure distribution that is physically consistent, and to account for any unknowable modeling errors by iterative regrasping.

(3) **The simplest pressure distribution, uniform pressure, is physically inconsistent.** If both contact patches have a uniform pressure distribution, then the two design variables available (the pressure at each patch) are insufficient to provide force-moment balance of the object in gravity.

(4) **The lowest-order physically consistent pressure distribution model is uniform pressure on finger 1 and linearly-varying pressure on finger 2.** The uniform pressure on finger 1 assures that the total normal force passes through the rotational joint above the finger. The linearly varying pressure distribution on finger 2 provides the extra variable needed to solve uniquely for the pressure distributions while assuring that the object remains in the plane of motion.

Figure 4.9 illustrates the two contact pressure distributions, viewed from the side. The pressure p_1 is constant over the contact patch, modeled as a disk of radius a. The pressure p_2 is also defined over a disk of radius a. Defining a y'-axis as the axis from the center of mass of the object to the center of the contact disk, the pressure distribution varies linearly along y' and is constant along the orthogonal direction (see Equation (B.3) in Appendix B for the expression of p_2). As shown in Figure 4.9, to maintain a static grasp, p_2 is larger closer to the center of mass and drops with increasing y'. This allows for force and moment balance considering the gravity force on the object.

Let γ be the distance from the object center of mass to the center of the finger contact patches. As shown in Appendix B, the minimum spring force S needed to maintain the grasp increases as γ increases, according to

$$S \geq \frac{f_g L}{L_S} \left(\frac{4\gamma}{a} - 1 \right). \tag{4.21}$$

For a spring force less than this bound, the required pressure p_2 for force-moment balance becomes negative within finger 2's contact patch, which is not physically realizable. Therefore moment balance cannot be achieved, and the object rotates and falls out of the plane of motion. You can try a simple experiment with a cell phone to see that a larger grip force is needed to hold the phone horizontal as the pinching fingers move further from the center of mass.

Based on the modeling above, Appendix B derives the detailed forms of the limit surfaces describing the contacts at finger 1 and finger 2. The limit surfaces are based on ellipsoidal approximations to the elliptic integrals corresponding to the finger contact forces and moments. The resulting closed-form expressions are $\mathbf{f}_i^{\mathsf{T}} \mathbf{A}_i \mathbf{f}_i = 1$, where for finger 1 $\mathbf{A}_1 = \mathbf{A}(0)$ and for finger 2 $\mathbf{A}_2 = \mathbf{A}(\gamma)$. The expression for $\mathbf{A}(\gamma)$ can be found in Equation (B.26) in Appendix B.

With the description of the contact limit surfaces as a function of the configuration of the object in the hand, we apply the dynamics and motion planning described in Sections 4.3–4.5 to experiments.

4.7. Experiment

We tested the motion planner discussed in Section 4.5 with a pinch-grasp introduced in Section 4.6 using the ERIN system described in Chapter 3.

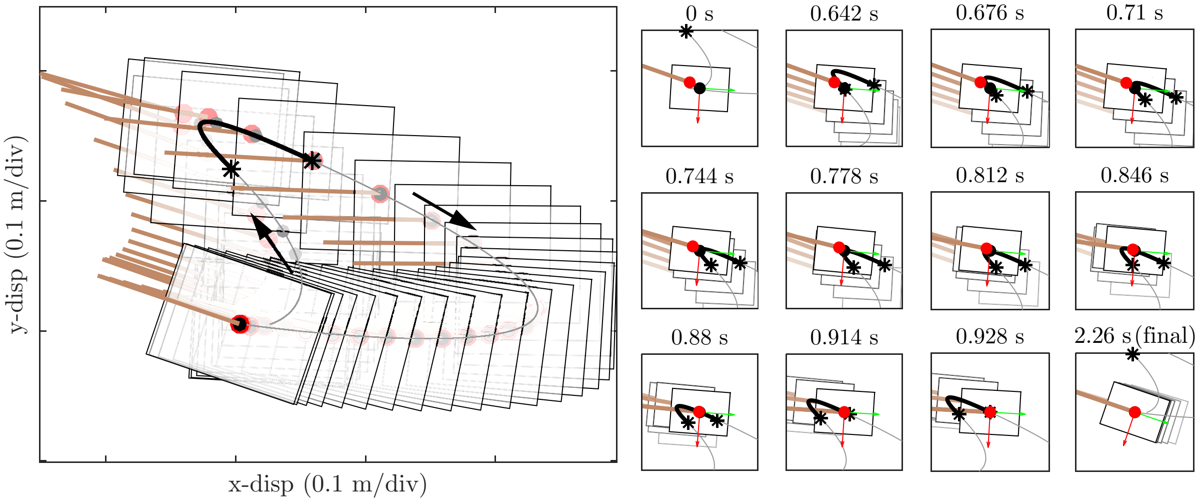


Figure 4.10. Repositioning example: showing trajectories found by the motion planner. The plot on the left shows the entire motion with a time interval between frames of 60 ms. Plots on the right show the initial and final configurations, and give more details of motion during the sliding phase. Solid gray curves are the object CM trajectories, red dots represent the finger contacts, and the brown lines represent the finger orientation. Thick black arrows show the directions of the object motion. Thick black curves show the sliding regions. Thin red and green arrows are the x and y directions of the body frame $\mathcal{B}$.

We used the lightweight passive gripper rather than the Allegro hand for the higher achievable accelerations and better contact force characterization as discussed in the beginning of Section 4.6. We used three joints of the 7-DOF WAM arm (joints 1, 4, and 6) while keeping the other joints fixed to emulate a 3R planar arm. Manipulator workspace, velocity, and acceleration constraints in the root-finding problem were based on joint, velocity, and motor properties given by the manufacturer as well as conservative estimates of the inertia matrix. The values of manipulator velocity and acceleration constraints are shown in Table 4.1.

Given initial and goal relative configurations, the motion planner plans the hand trajectory and solves the joint trajectories using inverse kinematics. The planned joint

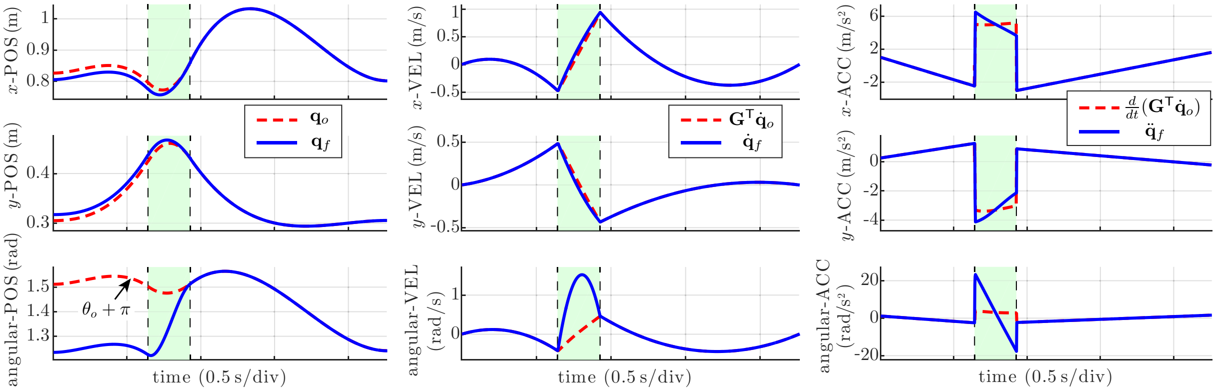


Figure 4.11. Repositioning example: showing the trajectories of the object (red dashed curves) and the finger (blue solid curves) found by the motion planner to reposition an object. Green shaded regions show the planned sliding phase. (Left) Finger contact center trajectories $\mathbf{q}_f$ and object CM trajectories $\mathbf{q}_o$. Initial relative position error is shown as the space between the dashed red line and the blue line which is reduced to zero after the sliding motion. (Middle) Finger velocities $\dot{\mathbf{q}}_f$ and object contact points velocities (points on the object that are coincident with the contact center) $\mathbf{G}^\mathsf{T}\dot{\mathbf{q}}_o$ are shown to demonstrate relative velocities at the contact. (Right) Finger accelerations $\ddot{\mathbf{q}}_f$ and object contact point accelerations $\frac{d}{dt}(\mathbf{G}^\mathsf{T}\dot{\mathbf{q}}_o)$ demonstrate relative accelerations at the contact.

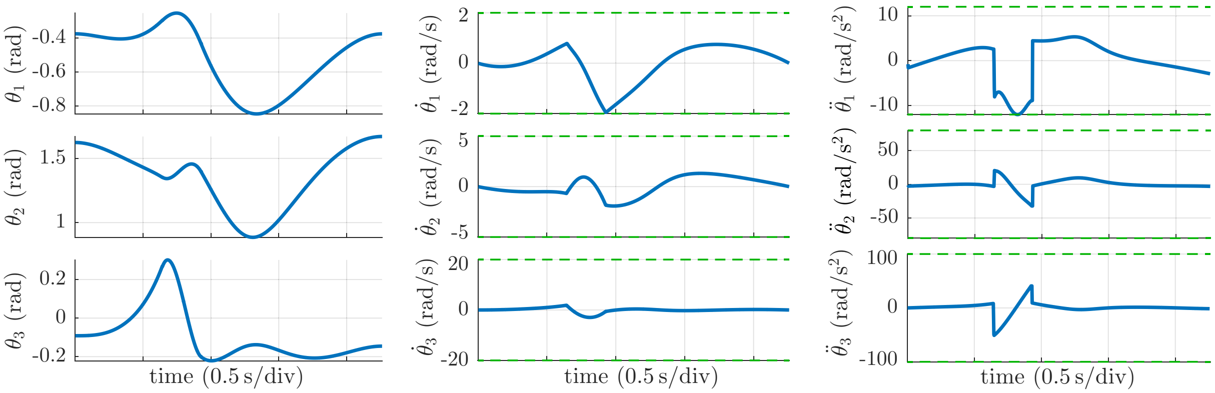
trajectories were generated offline, and real-time control was used to follow the trajectories specified by the planner. The motion control loop used encoder feedback and ran at 500 Hz on a Linux PC with an Intel Core i7-4770 CPU and 16 GB RAM. Motion was in a horizontal plane ($\alpha = 0$) to achieve more isotropic control authority than would be the case in a vertical plane. The values of the constants used in modeling and planning are summarized in Table 4.2.

joint #	$\dot{\mathbf{\Theta}}_{\min}$ (rad/s)	$\dot{\mathbf{\Theta}}_{\max}$ (rad/s)	$\ddot{\mathbf{\Theta}}_{\min}$ (rad/s^2)	$\ddot{\mathbf{\Theta}}_{\max}$ (rad/s^2)
1	-2	2	-12	12
4	-5	5	-80	80
6	-20	20	-100	100

Table 4.1. Limits for manipulator joint velocities and accelerations.

object mass m	0.049 kg
object dimensions	0.12 m×0.09 m
object inertia I (about its CM)	2.78×10^{-4} kg·m^2
angle between the horizontal plane α	0
gravity constant g	9.8 m/s^2
gripper arm length L	0.17 m
spring location L_S (as shown in Figure 4.9)	0.05 m
spring load S	-7 N
measured friction coefficient $\hat{\mu}$	0.34
friction coefficient used in planning μ	0.16
radius of the contact patch a	0.0254 m
3R robot link 1 length	0.552 m
3R robot link 2 length	0.303 m
3R robot link 3 length	0.287 m

Table 4.2. Parameter values for the two-fingered grasp system.



(a) Joint Positions (b) Joint Velocities (c) Joint Accelerations

Figure 4.12. Repositioning example: blue curves showing the planned joint positions, velocities, and accelerations of the manipulator calculated from the finger trajectories shown in Figure 4.11 by solving inverse kinematics. The green dashed lines are the joint velocity and acceleration limits corresponding to the values in Table 4.1. Joint position limits are not shown since the trajectories are far from the limits.

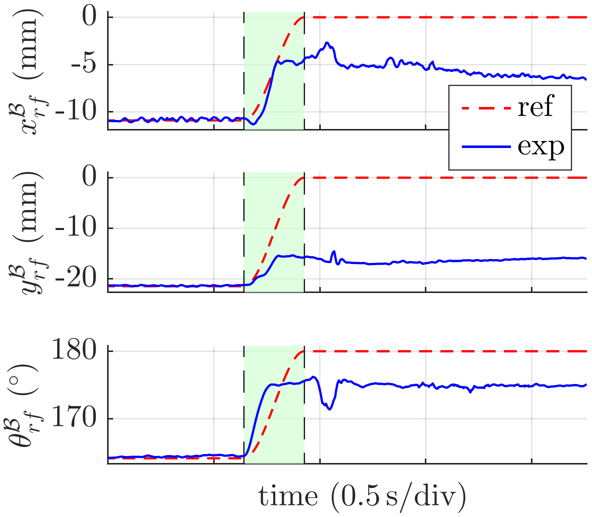


Figure 4.13. Experiment result of one-shot planning showing the relative position changes vs. time. Red dashed curves show the reference relative position $\mathbf{r}^{\mathcal{B}}_{f}$ trajectories. Blue curves represent the actual relative position trajectories. Green shaded regions indicate the planned sliding phase.

4.7.1. One-shot Planning and Execution

Before planning, the initial relative position $\mathbf{r}^{\mathcal{B}}_{f,\text{init}}$ was measured by the vision system. With the goal relative position $\mathbf{r}^{\mathcal{B}}_{f,\text{goal}}$ given by the user, the motion planner calculated the motion of the robot to realize the repositioning satisfying all the constraints.

Figures 4.10 and 4.11 show the motion planning result of a sliding regrasp example. The initial relative position was measured as $\mathbf{r}^{\mathcal{B}}_{f,\text{init}} = [-0.011\,\text{m},\ -0.022\,\text{m},\ 164°]^{\mathsf{T}}$, and the goal relative position was given as $\mathbf{r}^{\mathcal{B}}_{f,\text{goal}} = [0, 0, 180°]^{\mathsf{T}}$. Figure 4.12 shows the planned joint trajectories of the 3R robot which were calculated from the inverse kinematics applied to the finger position trajectories in Figure 4.11. The planned time periods for each phase were $T_1 = 0.64\,\text{s}$, $T_2 = 0.29\,\text{s}$ and $T_3 = 1.33\,\text{s}$.

The planned motions were tested experimentally. The WAM arm followed the pre-planned joint trajectories using a PID-based joint position controller. The object poses were obtained from the vision system. To prevent overshoot during the sliding motion, we

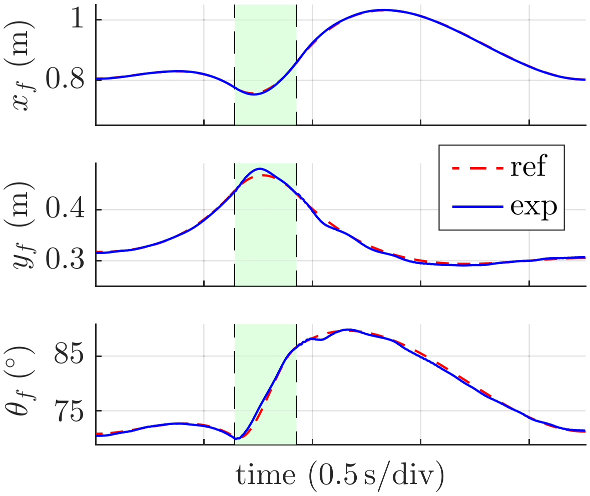


Figure 4.14. Experiment result of one-shot planning, showing the finger contact center trajectories. Green shaded regions indicate the planned sliding phase. The mean absolute tracking errors are $[1.02\,\text{mm}, 2.56\,\text{mm}, 0.296\,°]^\mathsf{T}$, and the standard deviations are $[1.04\,\text{mm}, 2.75\,\text{mm}, 0.27\,°]^\mathsf{T}$.

chose an underestimated friction coefficient in the motion planner. Experimental results of relative position change are shown in Figure 4.13. Finger position tracking results are shown in Figure 4.14, where the finger poses were calculated from the recorded joint angles and the forward kinematics of the system.

During the implementation, the finger moved relative to the object along the desired direction and ended up with some undershoot errors, which were more apparent in the y-direction. One reason for this undershoot is the intentionally underestimated friction coefficient in the motion planner. Another reason is the trajectory tracking error is larger in the y-direction as shown in Figure 4.14. Since there is more error in the y-direction from the initial configuration, the planned motion has higher accelerations in this direction which increases tracking error. The final relative position errors could also have been caused by modeling errors and uncertainties in measuring the initial relative positions.

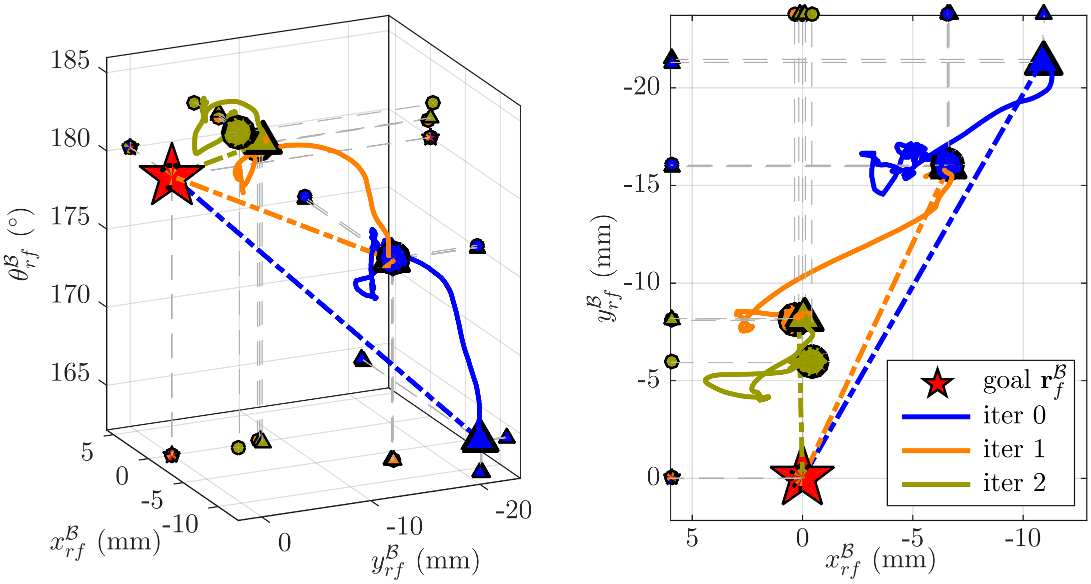
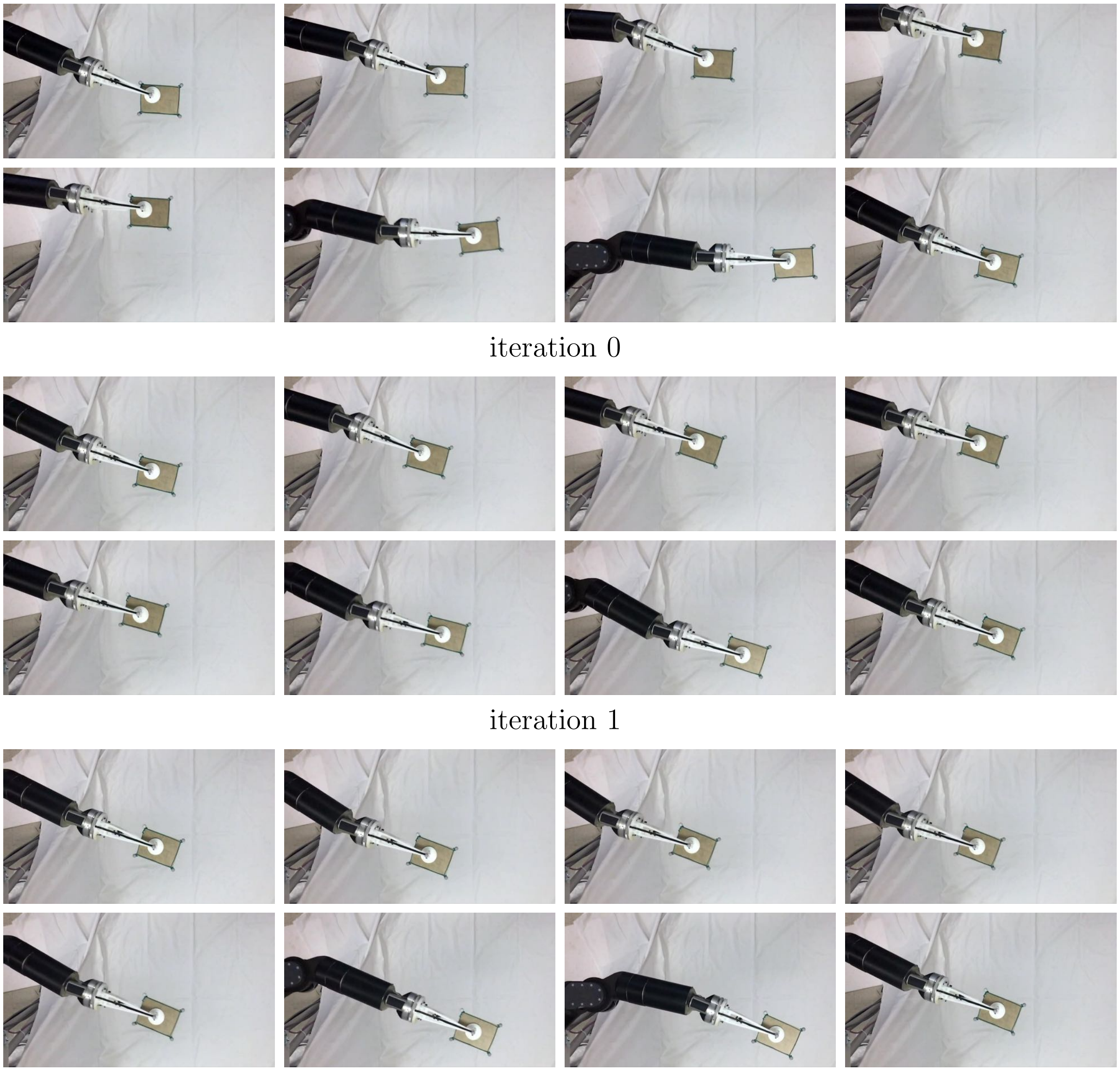


Figure 4.15. Experimental results of iterative planning and execution for one experiment consisting of three iterations. The plots show the planned and actual relative configurations $\mathbf{r}^{\mathcal{B}}_{f}(t)$. Each color represents one iteration. Triangles and circles show the initial and final points of each trajectory. The dashed lines are the planned trajectories and the solid curves shows the experimental results. Plots on the left and right show the same result from different viewpoints.

4.7.2. Iterative Planning and Execution

This section reports the results of iterative planning and execution for 3-DOF planar regrasping. Unlike the idealized 1-DOF example in Section III, we have no theoretical convergence results for iterative in-hand regrasping for the 3-DOF case and all possible sources of error. The motivation for iterative regrasping (essentially discrete-time one-step deadbeat feedback control) is the same, however.

In each experiment, we tested three iterations of motion planning and execution with the same goal. In each iteration, the initial state was measured from the last state of the previous iteration. The robot trajectories were planned automatically by the motion



iteration 0

iteration 1

iteration 2

Figure 4.16. Iterative planning experiment corresponding to Figure 4.15. Total times for iterations 0, 1, and 2 are 2.26 s, 1.6 s, and 1.95 s respectively. Time intervals between snapshots were manually chosen to show the motion of the system. A video of this experiment is shown in the attached media.

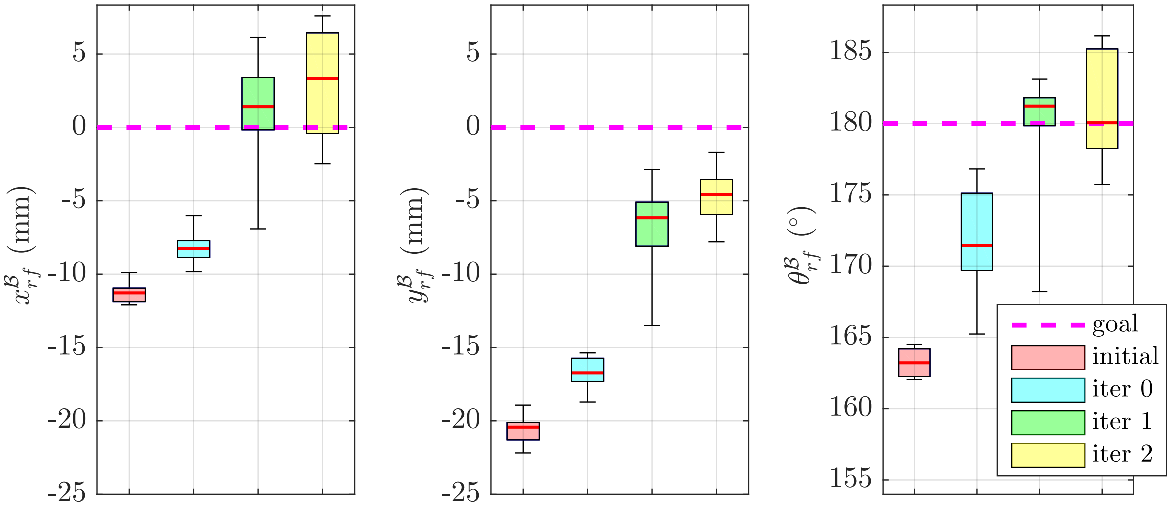


Figure 4.17. Experiment results of 10 trials, showing the changes of relative positions. Each color represents one iteration. The red lines within the boxes show the mean values, edges of the boxes are the 25th and 75th percentiles, and whiskers extend to the most extreme data points.

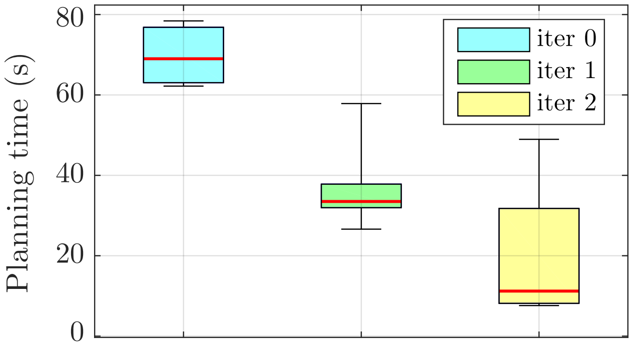


Figure 4.18. Planning time of 10 trials. Each color represents one iteration. Red lines within the boxes show the mean values, edges of the boxes are the 25th and 75th percentiles, and whiskers extend to the most extreme data points.

planner based on the initial state and goal state. Then the robot followed the planned trajectories.

Results of the iterative planning are shown in Figures 4.15 and 4.16. The first iteration corresponds to the example given in Section 4.7.1. After each iteration, the relative position was closer to the goal configuration. Once the object is near the goal state, additional iterations did not decrease the error. In cases where the error was close to the mean vision error of the vision system ($\sim 0.5\,\mathrm{mm}$), additional iterations could actually introduce more error.

4.7.3. Repeatability

To test the repeatability of the repositioning experiment, we ran the previous three-iteration experiment ten times, making a total of 30 motion plans and executions. At the beginning of each three-iteration trial, the object was manually placed at approximately the same initial configuration. Figure 4.17 shows a boxplot of the relative position changes. Figure 4.18 shows a boxplot of the planning time. Further iterations produce no statistically significant improvement (or worsening) of the grasp.

Sources of error in achieving planned regrasps include vision errors (as mentioned above), error in following the planned robot trajectory, and contact modeling errors (e.g., the Coulomb friction approximation and the contact pressure distribution approximations). The mean absolute robot trajectory tracking errors were $[1.2\,\mathrm{mm}, 2.6\,\mathrm{mm}, 0.35\,°]^\mathsf{T}$, with standard deviations $[0.91\,\mathrm{mm}, 2.6\,\mathrm{mm}, 0.22\,°]^\mathsf{T}$, respectively. Errors induced by the assumed form of the contact pressure distributions are likely less meaningful for n-fingered grasps than for our 2-fingered grasp, because the distances between the finger contacts play a larger role in determining the shape of the grasp limit surface than the detailed pressure distribution at single fingers. While our finger/object contacts approximately obeyed a dry Coulomb friction model, other finger/object contacts, particularly involving soft, hysteretic materials such as rubber, may require a different contact model.

The goal configuration was the same in all 30 individual planning and execution steps, but the initial configurations varied considerably. The results of the reported experiments are in line with what we observed with other 3-DOF 2-fingered laminar regrasps we tried, and point to a typical final error on the order of a few millimeters in linear position and a few degrees in orientation. Achieving a smaller final error would require improvements in the vision system, trajectory tracking, or contact modeling.

CHAPTER 5

In-hand Manipulation with Spring-Sliding Compliance

In the previous chapter, an object is regrasped using dynamic loads and sliding is restricted to parallel surfaces. In this work we focus on quasistatic regrasping by pushing theobject against stationary constraints. Each finger is modeled as a frictional point contact connected by a three-dimensional linear spring to an anchor point whose motions is controlled in three linear directions. Given the finger stiffness, by position controlling the anchor we can change the fingertip contact force and initiate sliding when the contact forces are at the edges of their friction cones. External contacts provide forces that maintain object force balance during the motions. More details can be found in Section 5.2.

Similar to spring compliance that maps contact forces to finger compressions, sliding can be treated as another source of compliance that yields accommodating sliding directions. Advantages of spring-sliding compliant grasps include:

- The spring compliance ensures that fingers remain in contact while sliding over general surfaces in addition to parallel planes.
- With spring compliance the contact forces are determined by finger compressions, so contact force control can be achieved by controlling finger anchor motions.
- Sliding compliance bounds the tangential contact forces and allows sliding motions for in-hand manipulation.
- A spring-compliant grasp can be both programmable and mechanically passive, ensuring stability [**24**], [**28**].

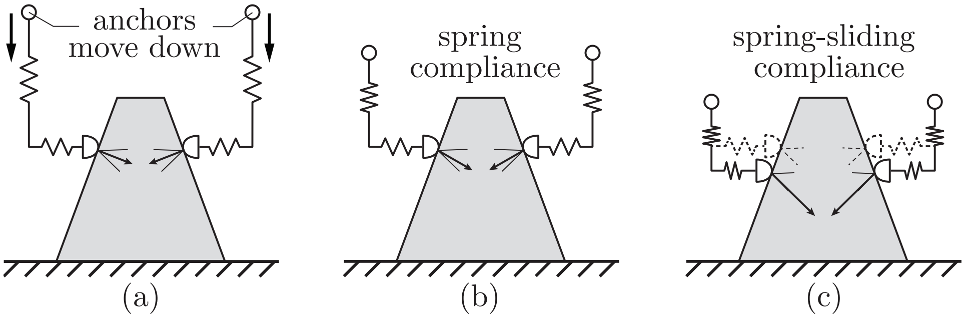


Figure 5.1. Example showing spring-sliding compliance: two springy fingers grasp an object with two point contacts. The spring anchors move down vertically and push the object against a fixed table. Lines at the contact points show friction cones and arrows show the contact forces applied to the object. When the anchors move from (a) to (b), the fingers are further compressed and the contact forces increase in the vertical direction. The fingertips are still sticking since the contact forces are within the friction cones. As the anchors move to (c), the contact forces reach the edges of the friction cones and the fingertips start to slide on the contact surfaces. Contact forces from the table keep the object stationary and force-balanced.

We define a grasp configuration as a set of finger relative contact positions to the object. The goal is to achieve a desired new grasp by controlling the anchor motions while maintaining object quasistatic force balance. Figure 5.1 shows an example of in-hand regrasping. When the anchors move down, fingertip contact forces are first moved to the edges of the friction cones and then sliding is initiated to achieve regrasping. The object remains stationary and contacts with the environment provide external forces to keep the object force-balanced.

In general case, we assume the external contacts locations are fixed in the object frame and the external environment is stationary. This allows rolling and sliding motions of the object with respect to the environment other than remain stationary. The problem that we address in this paper is to find the object and anchor motions that can realize a given regrasp for multi-fingered point-contact grasps with environment contacts. A more complete problem statement and list of assumptions are given in Section 5.1.

The main results and contributions can be summarized as following:

(1) *Contact mechanics of a single finger* (Section 5.3) : Given a system configuration (including object, finger anchor, and contact point positions) and the object velocity, find the relationship between the anchor velocities and the fingertip velocities.

(2) *Object mechanics* (Section 5.4) : Given the motions of the object and all the anchors, find object velocity constraints that maintain external contacts and force-balance constraint in terms of the forces from the fingers and the external contacts.

(3) *Robustness analysis* (Section 5.5) : A set of object and anchor trajectories are robust when finger contact wrench uncertainties can always be balanced by counter wrenches from the environment contacts. The goal of the analysis is to find the robust conditions for a given set of object and anchor motions with respect to given uncertainties.

(4) *Motion planning* (Section 5.6) : Find feasible and robust object and finger anchor trajectories that can realize a desired regrasp satisfying constraints derived from (2).

(5) *Implementation* (Section 5.7) : We detail a two-fingered system and specify the motion planning algorithm in (4) to find feasible anchor trajectories that can realize a desired regrasp and maximize the robustness assuming the object's motion remains stationary. Simulation and experiment results are presented for a planar regrasp example.

5.1. Problem Description

The system consists of an n-fingered hand grasping an object with n point contacts. Each finger anchor can be motion-controlled individually. The object also contacts a rigid stationary environment with a total of m frictional point contacts (a line contact is approximated by two points and a face contact by three or more points). A system configuration is defined by the positions of the finger anchors, finger contact points, and the object. The problem can be described as: given an initial system configuration where the object is force balanced with no relative motion at the fingertips and a desired new grasp, find anchor and object motions that realize the regrasp while maintaining force balance at all times.

5.1.1. Assumptions

(1) Quasistatic motions.

(2) Fingers contact the object at point fingertips.

(3) Each finger is linearly springy and the stiffness is known. The stiffness matrix is symmetric positive definite to ensure grasp stability [**24, 3, 28, 27**], i.e., for small position disturbance of the object the grasp is able to restore the object to its desired location instead of tending to eject the object.

(4) Each finger maintains a positive contact normal force.

(5) The object is rigid, smooth, and of known geometry.

(6) Dry Coulomb friction applies at each contact. The friction coefficients at all contacts are known. For convenience, we assume that finger contacts with the object have a friction coefficient μ and environment contacts with the object have a friction coefficient μ_e.

(7) The m external contact points are given and fixed w.r.t. the object.

5.1.2. Notations

Within the scope of this paper, all vectors are written in bold lowercase letters, all matrices are written in bold capital letters, scalars are written in italic letters and coordinate frames

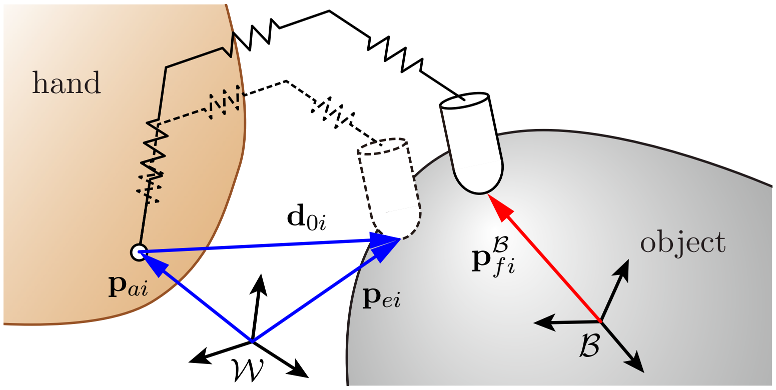


Figure 5.2. System configuration.

are denoted with calligraphic letters. All variables are in the world frame $\mathcal{W}$ unless noted otherwise in the superscripts. For example, $\mathbf{p}_{fi}$ is fingertip position of the ith finger in the world frame $\mathcal{W}$ and $\mathbf{p}_{fi}^{\mathcal{B}}$ is the fingertip position in the object frame $\mathcal{B}$.

1) For the object:

$\mathcal{B}$ Object frame attached to the object.

$\mathbf{p}_o$ The position of the origin of $\mathcal{B}$, $\mathbf{p}_o = [x_o, y_o, z_o]^T$.

$\mathbf{R}_o$ Rotation matrix of the object, $\mathbf{R}_o \in SO(3)$.

$\boldsymbol{\omega}_o$ Object angular velocity. $[\boldsymbol{\omega}_o] \in so(3)$ is the skew-symmetric matrix representation of $\boldsymbol{\omega}_o$.

2) For the fingers:

$\mathbf{p}_{fi}$ The ith fingertip position, $\mathbf{p}_{fi} = [x_{fi}, y_{fi}, z_{fi}]^T$.

$\mathbf{p}_{ai}$ The ith anchor position, $\mathbf{p}_{ai} = [x_{ai}, y_{ai}, z_{ai}]^T$.

$\mathbf{d}_{0i}$ The equilibrium position of the ith fingertip.

$\mathbf{d}_i$ Compression of the ith finger, $\mathbf{d}_i = \mathbf{p}_{fi} - \mathbf{d}_{0i} - \mathbf{p}_{ai}$.

$\mathbf{K}_i$ Stiffness matrix of the ith finger, $\mathbf{K}_i \in \mathbb{R}^{3\times 3}$.

Figure 5.2 shows the configuration of the n-fingered grasp for a general spatial case.

3) For the contact forces:

The contact force applied to the object by the ith finger is

$$\mathbf{f}_{ci} = -\mathbf{K}_i \mathbf{d}_i = -\mathbf{K}_i(\mathbf{p}_{fi} - \mathbf{p}_{ai} - \mathbf{d}_{0i}). \tag{5.1}$$

With the given contact surface geometry, the contact normal direction which always points into the object is a function of the finger contact position in $\mathcal{B}$

$$\hat{\mathbf{n}}_i(\mathbf{p}_{fi}^{\mathcal{B}}) = \mathbf{R}_{fi}[0, 0, 1]^T, \tag{5.2}$$

where the hat means the vectors are normalized. The contact normal force is the projection of $\mathbf{f}_{ci}$ to the normal direction as

$$\mathbf{f}_{Ni} = (\mathbf{f}_{ci} \cdot \hat{\mathbf{n}}_i)\hat{\mathbf{n}}_i = \mathbf{f}_{ci}^T \hat{\mathbf{n}}_i \hat{\mathbf{n}}_i, \tag{5.3}$$

and the contact tangential force is

$$\mathbf{f}_{ti} = \mathbf{f}_{ci} - \mathbf{f}_{Ni}. \tag{5.4}$$

5.2. Finger Spring Compliance Model

In this paper we introduce a general finger spring compliance model in order to realize sliding control through controlling the finger anchor positions. In this section we show how this model can represent physical fingers. Figure 5.3 shows two different type of fingers. For (a), there is a spring-mounted fingertip attached to the end of a position-controlled finger. The anchor point is the end of the position-controlled finger. This design directly matches the model as long as the 3D stiffness of the spring mounting is known and the 3D spring deflection is measured.

Figure 5.3(b) represents the case where the fingertip is rigidly mounted to the finger. The effective stiffness may come from an active stiffness control law or from passive compliance at the joints (as with series elastic actuators) or at the links. Another interesting

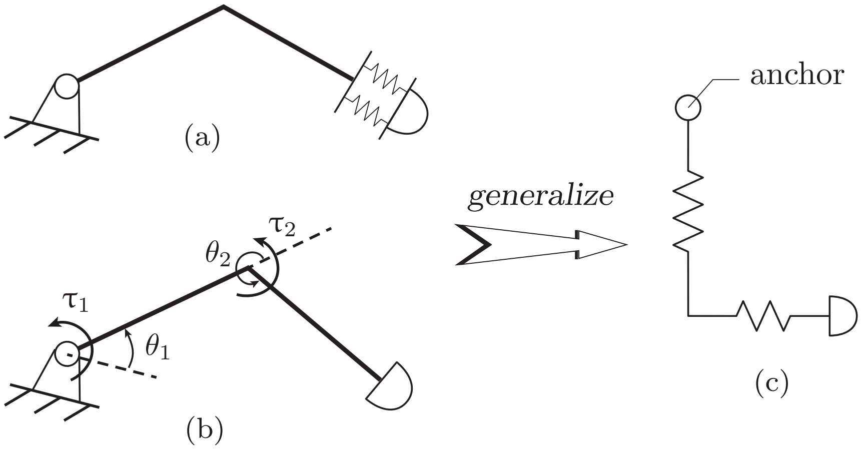


Figure 5.3. Generalized finger spring compliance model. (a) shows a compliant unit fixed to the fingertip of an otherwise rigid position-controlled finger. (b) shows a joint-torque controlled finger. Fingertip stiffnesses of both cases can be modeled by (c).

case is where there is no passive compliance but the joints are torque controlled. In this case, the anchor is the base of the finger and the entire finger acts as a spring. We examine this case in more detail below.

Let $\boldsymbol{\theta}$ denote the finger joint angle vector, $\boldsymbol{\tau}$ denote the joint torque vector, and $\mathbf{J}(\boldsymbol{\theta})$ denote the Jacobian matrix. From finger kinematics and using the principal of virtual work, we have the mapping from fingertip contact forces to the joint torques $\boldsymbol{\tau} = \mathbf{J}^T\mathbf{f}_c$. When $\mathbf{J}$ is invertible, we have

$$\begin{aligned} &\mathbf{f}_c = \mathbf{J}^{-T}\boldsymbol{\tau} \\ &\rightarrow \partial\mathbf{f}_c = \partial(\mathbf{J}^{-T})\boldsymbol{\tau} + \mathbf{J}^{-T}\partial\boldsymbol{\tau}. \end{aligned} \tag{5.5}$$

For simplicity we assume the world frame is at the finger base and $\mathbf{p}_f$ is the fingertip position relative to the anchor. From the velocity mapping $\dot{\mathbf{p}}_f = \mathbf{J}\dot{\boldsymbol{\theta}}$ we have

$$\partial\mathbf{p}_f = \mathbf{J}\partial\boldsymbol{\theta}. \tag{5.6}$$

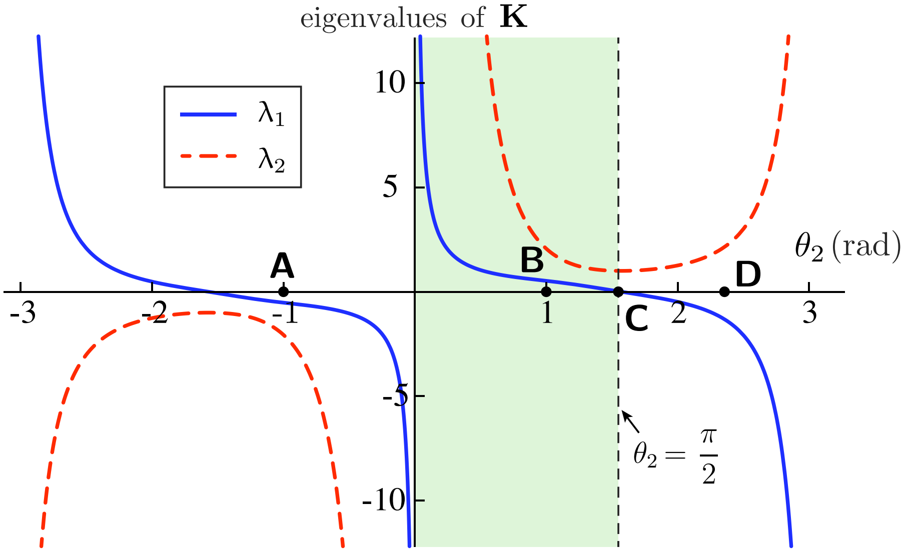


Figure 5.4. Eigenvalues of stiffness matrix $\mathbf{K}$ for the example 2R finger. In this example, $\tau_1 = \tau_2 = 1\,\mathrm{Nm}$ and both link lengths are 1 m. The green shaded region shows the feasible range for θ_2 that can yield positive definite $\mathbf{K}$.

Combining Equations (5.5) and (5.6), we can write the finger stiffness matrix as

$$\mathbf{K} = -\frac{\partial \mathbf{f}_c}{\partial \mathbf{p}_f} = -\frac{\partial(\mathbf{J}^{-T})}{\partial \boldsymbol{\theta}}\boldsymbol{\tau}\mathbf{J}^{-1} - \mathbf{J}^{-T}\frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{\theta}}\mathbf{J}^{-1}. \tag{5.7}$$

The specific expression for $\mathbf{K}$ depends on the Jacobian and how the joint torques $\boldsymbol{\tau}$ are commanded.

Assume, for example, that joint torques are independent of the finger position ($\frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{\theta}} = \mathbf{0}$), i.e., open-loop torque control. As an example, consider a 2R finger example as shown in Figure 5.3(b). We assume that the links have unit length and the joint torques have a constant value of 1. Then the stiffness matrix simplifies to

$$\mathbf{K}(\boldsymbol{\theta}) = -\frac{\partial(\mathbf{J}^{-T})}{\partial \boldsymbol{\theta}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\mathbf{J}^{-1} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}, \tag{5.8}$$

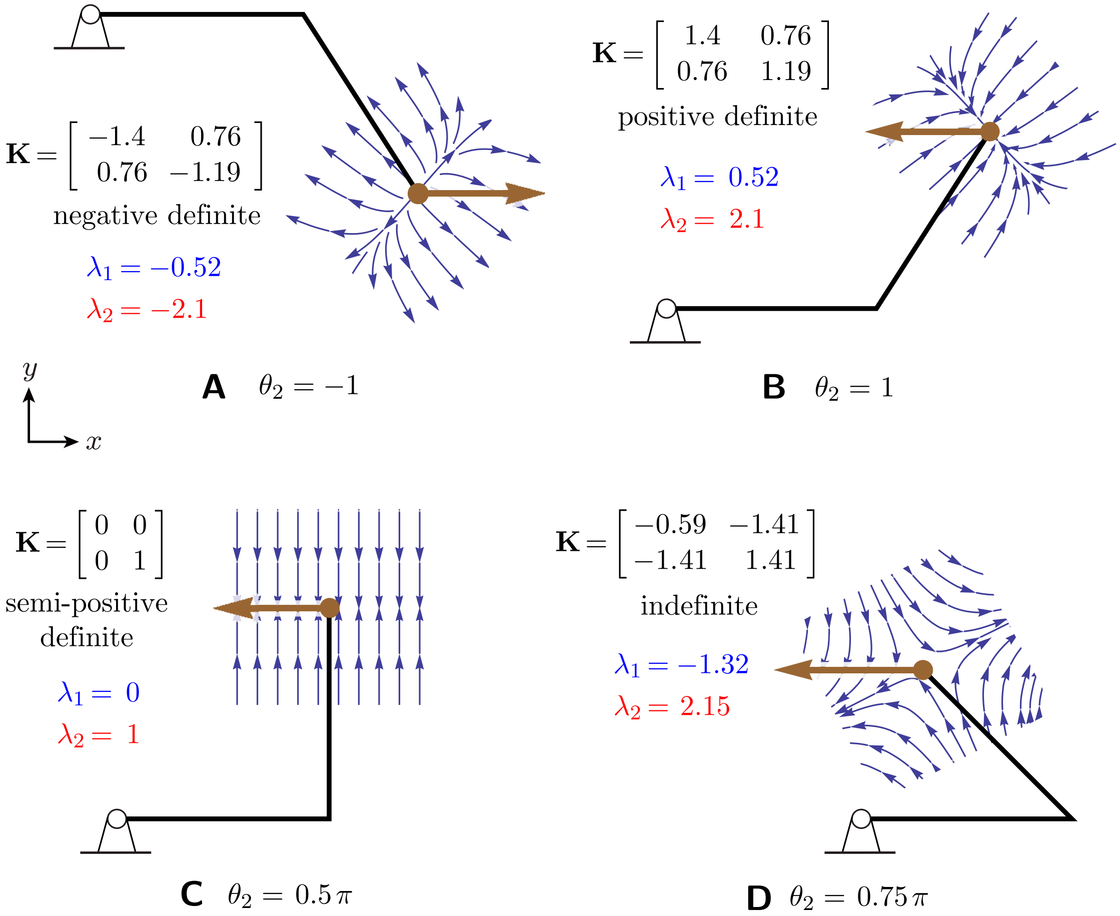


Figure 5.5. We pick some points on Figure 5.4 and draw corresponding finger configurations with $\theta_1 = 0$. Each chosen θ_2 and corresponding stiffness matrix $\mathbf{K}$ and its eigenvalues by Equation (5.8) are shown. Dashed lines show eigenvectors of $\mathbf{K}$. Stream plots at fingertips show how contact force change $\partial \mathbf{f}_c$ reacting to fingertip position changes $\partial \mathbf{p}_f$. Brown arrows represent fingertip contact forces.

where

$$k_{11} = \frac{1}{4} \csc^3 \theta_2 \left(\cos\left(2\theta_1 - \theta_2\right) + 2\left(\cos \theta_2 + \cos\left(2\theta_1 + 2\theta_2\right) + 1\right) + \cos\left(2\theta_1 + \theta_2\right) \right),$$

$$k_{12} = k_{21} = \frac{1}{4} \left(\sin\left(2\theta_1 - \theta_2\right) + 2 \sin\left(2\theta_1 + 2\theta_2\right) + \sin\left(2\theta_1 + \theta_2\right) \right) \csc^3 \theta_2,$$

$$k_{22} = -\frac{1}{4} \csc^3 \theta_2 \left(\cos\left(2\theta_1 - \theta_2\right) - 2(\cos \theta_2 - \cos\left(2\theta_1 + 2\theta_2\right) + 1) + \cos\left(2\theta_1 + \theta_2\right) \right).$$

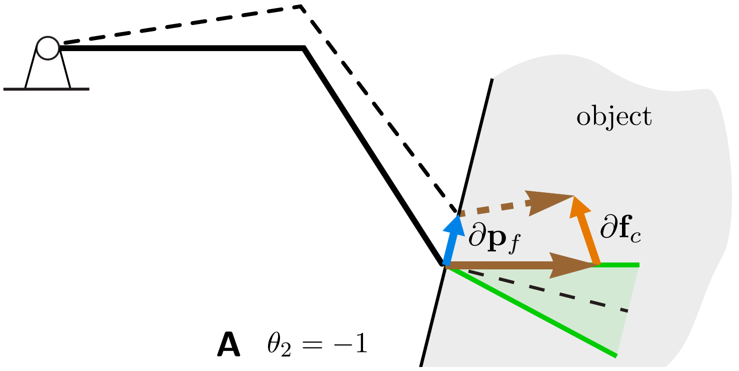


Figure 5.6. An unstable sliding example for case **A** in Figure 5.5: since $\tau_1 = \tau_2$ the fingertip force always aligns with the first link. For a fingertip displacement $\partial \mathbf{p}_f$ as the blue vector, the corresponding contact force change $\partial \mathbf{f}_c$ shown as the orange vector will keep rotating counterclockwise and remain outside the friction cone. The green shaded area shows the friction cone.

The eigenvalues of $\mathbf{K}$ are

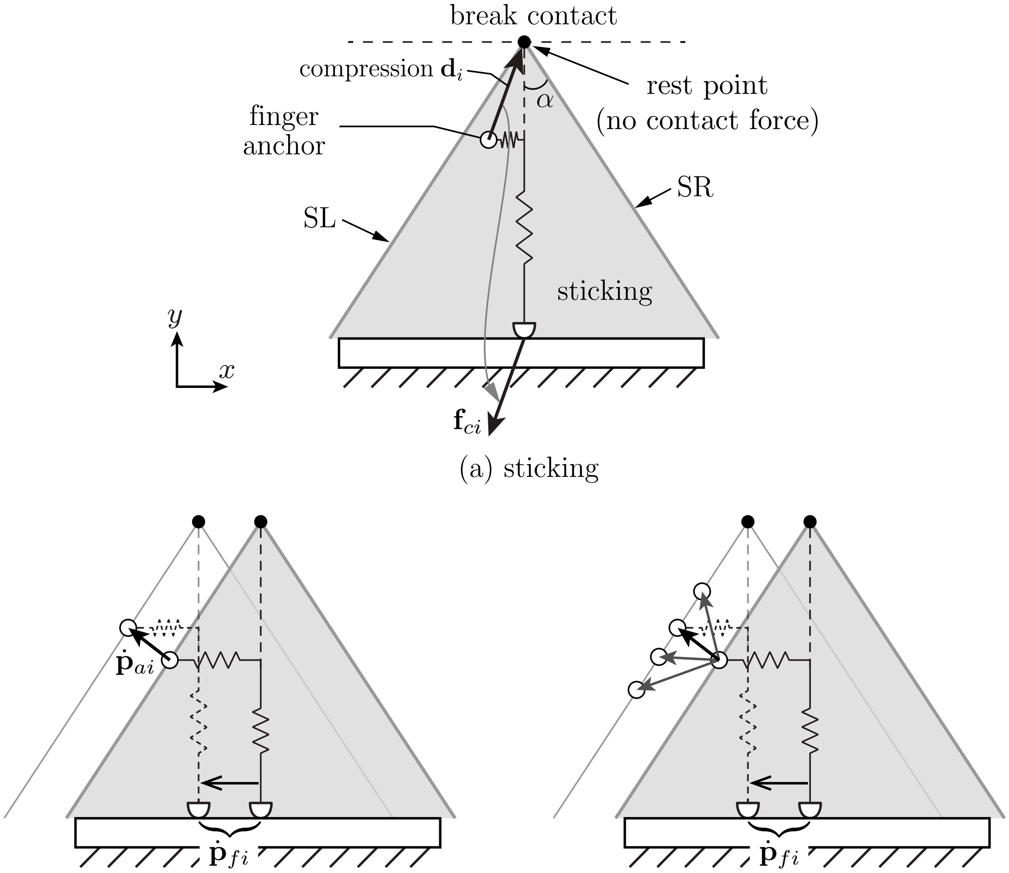
$$\lambda_1 = \frac{1}{2}\csc^3\theta_2\left(1+\cos\theta_2 - \sqrt{1+\cos(3\theta_2)+\cos\theta_2+\cos^2\theta_2}\right),$$
$$\lambda_2 = \frac{1}{2}\csc^3\theta_2\left(1+\cos\theta_2 + \sqrt{1+\cos(3\theta_2)+\cos\theta_2+\cos^2\theta_2}\right).$$

The eigenvalues are only related to θ_2 since θ_1 only changes the finger's orientation relative to the base. The stiffness matrix $\mathbf{K}$ is symmetric and the two eigenvalues must both be positive to satisfy the assumption of positive-definite stiffness. We plot the eigenvalues with respect to θ_2 as shown in Figure 5.4. And Figure 5.5 shows the finger configuration and stiffness for four values of θ_2. From the plots the finger configuration should satisfy $0 < \theta_2 < \frac{\pi}{2}$ to satisfy the positive-definite stiffness assumption. In Figure 5.5, **A**, **C** and **D** are cases where the stiffness matrix is not positive definite. When the fingertip slides on the contact surface, these cases may cause unstable sliding conditions. In Figure 5.6 we show an example for case **A** of Figure 5.5 to illustrate the unstable sliding and violation of the quasistatic assumption: given a line contact with an object and the friction cone as shown, initially the contact force lies on the edge of the friction cone. By either moving the anchor or the object, for a fingertip position change $\partial \mathbf{p}_f$ as the blue vector, the corresponding contact force change $\partial \mathbf{f}_c$ is shown as the orange vector. The contact force goes out of the friction cone and initiates sliding. In this example since $\boldsymbol{\tau}_1 = \boldsymbol{\tau}_2$ the fingertip force always aligns with the first link. As the fingertip moves upwards, the contact force will keep rotating counterclockwise and remain outside the friction cone. Therefore the fingertip will keep sliding and never reach an equilibrium state. We provide more analysis of the unstable sliding conditions for general cases in Section 5.3.4.

5.3. Contact Mechanics of A Single Finger

This section answers the following question: given the ith finger anchor and contact positions, and the object position and velocity, what is the relationship between the finger anchor velocities $\dot{\mathbf{p}}_{ai}$ and the corresponding fingertip motions $\dot{\mathbf{p}}_{fi}$?

First we need to tell which contact mode each fingertip is in. For the point contacts of the fingertips, the possible contact modes are sticking or sliding. Given the anchor and contact locations, the contact forces can be determined by the spring compliance. When the contact forces are in the interior of the friction cones, the fingers stick on the object and the contact points follow the motion of the object. When the fingertips slide the contact forces are on the boundaries of the friction cones and the fingertip motions are aligned with the tangential contact forces. For the sliding case, the *forward mechanics* problem is to find the contact point velocity $\dot{\mathbf{p}}_{fi}$ given the anchor velocity $\dot{\mathbf{p}}_{ai}$, and the *inverse mechanics* problem is to find the anchor velocity $\dot{\mathbf{p}}_{ai}$ corresponding to a given desired contact sliding velocity $\dot{\mathbf{p}}_{fi}$. A simple example of a planar 1-finger system is shown in Figure 5.7. Although in this example the forward and inverse mechanics can be solved geometrically, below we analytically derive the explicit solutions for the general spatial case.



(a) sticking

(b) sliding (forward)

(c) sliding (inverse)

Figure 5.7. A planar 1-finger example: the object is fixed and the finger stiffness matrix is the identity matrix, therefore the contact force is $\mathbf{f}_{ci} = -\mathbf{d}_i$. The finger rest point is right above the contact point. If the anchor moves above it the finger will break contact. The shaded area is the sticking region for the anchor positions, $\alpha = \tan^{-1}\mu$. (a) shows a sticking case. (b) shows a sliding case of the forward problem: given an anchor velocity $\dot{\mathbf{p}}_{ai}$, find the finger contact point velocity $\dot{\mathbf{p}}_{fi}$. When the anchor is at the edge of the sticking region, the motion $\dot{\mathbf{p}}_{ai}$ "pushes" the sticking region and causes the contact to slide left. And (c) shows an inverse problem example: all the anchor motions that go to the left edge of the new sticking region yield to the same sliding motion $\dot{\mathbf{p}}_{fi}$.

5.3.1. Sticking Case

When sticking the ith contact follows the object, i.e.,

$$\dot{\mathbf{p}}_{fi}^{\mathcal{B}} = \mathbf{0}^{3\times1}. \tag{5.9}$$

The transformations of the contact position and velocity from $\mathcal{B}$ to $\mathcal{W}$ can be written as

$$\mathbf{p}_{fi} = \mathbf{p}_o + \mathbf{R}_o \mathbf{p}_{fi}^{\mathcal{B}}\ , \tag{5.10}$$

$$\dot{\mathbf{p}}_{fi} = \dot{\mathbf{p}}_o + \dot{\mathbf{R}}_o \mathbf{p}_{fi}^{\mathcal{B}} + \mathbf{R}_o \dot{\mathbf{p}}_{fi}^{\mathcal{B}}. \tag{5.11}$$

Substituting Equation (5.9) into Equation (5.11), we have the contact velocity in the sticking case as

$$\dot{\mathbf{p}}_{fi} = \dot{\mathbf{p}}_o + \dot{\mathbf{R}}_o \mathbf{p}_{fi}^{\mathcal{B}} = \dot{\mathbf{p}}_o + [\boldsymbol{\omega}_o]\mathbf{R}_o \mathbf{p}_{fi}^{\mathcal{B}}. \tag{5.12}$$

5.3.2. Sliding Case – Forward Mechanics

In sliding cases we first solve the forward problem as finding the corresponding $\dot{\mathbf{p}}_{fi}$ for a given anchor velocity $\dot{\mathbf{p}}_{ai}$. When sliding the contact forces of the ith finger satisfy

$$\|\mathbf{f}_{ti}\| = \mu\|\mathbf{f}_{Ni}\|. \tag{5.13}$$

The sliding direction is along the tangential force direction

$$\hat{\dot{\mathbf{p}}}_{fi} = \hat{\mathbf{f}}_{ti}. \tag{5.14}$$

We define the finger sliding velocity w.r.t. $\mathcal{B}$ as

$$\dot{\mathbf{p}}_{fi}^{\mathcal{B}} = \lambda_i \mathbf{f}_{ti}^{\mathcal{B}} = \lambda_i \mathbf{R}_o^T \mathbf{f}_{ti}, \tag{5.15}$$

where λ_i is a positive scalar showing how fast the fingertip slides. During sliding the contact force of the fingertip remains on the boundary of the friction cone. We use this relationship to solve for λ_i as a function of the given system configuration and object velocity:

First, substituting Equation (5.15) into (5.11), we have

$$\begin{aligned} \dot{\mathbf{p}}_{fi} &= \dot{\mathbf{p}}_o + \dot{\mathbf{R}}_o \mathbf{p}_{fi}^{\mathcal{B}} + \mathbf{R}_o \lambda_i \mathbf{R}_o^T \mathbf{f}_{ti} \\ &= \mathbf{c}_{fi} + \lambda_i \mathbf{f}_{ti}, \end{aligned} \tag{5.16}$$

where $\mathbf{c}_{fi} = \dot{\mathbf{p}}_o + [\boldsymbol{\omega}_o]\mathbf{R}_o \mathbf{p}_{fi}^{\mathcal{B}}$ reflects the change of the contact point position due to the object motion.

From Equation (5.13), we can derive that

$$\begin{aligned} & \|\mathbf{f}_{ti}\|\|\mathbf{f}_{ti}\| = \mu^2 \|\mathbf{f}_{Ni}\|\|\mathbf{f}_{Ni}\| \\ & \rightarrow \mathbf{f}_{ti} \cdot \mathbf{f}_{ti} = \mu^2\, \mathbf{f}_{Ni} \cdot \mathbf{f}_{Ni} \\ & \xrightarrow{\frac{d}{dt}} \dot{\mathbf{f}}_{ti} \cdot \mathbf{f}_{ti} + \mathbf{f}_{ti} \cdot \dot{\mathbf{f}}_{ti} = \mu^2 (\dot{\mathbf{f}}_{Ni} \cdot \mathbf{f}_{Ni} + \mathbf{f}_{Ni} \cdot \dot{\mathbf{f}}_{Ni}) \\ & \rightarrow \mathbf{f}_{ti}^T \dot{\mathbf{f}}_{ti} = \mu^2\, \mathbf{f}_{Ni}^T \dot{\mathbf{f}}_{Ni}. \end{aligned} \tag{5.17}$$

Then from Equation (5.2) we have

$$\dot{\hat{\mathbf{n}}}_i = \frac{\partial \hat{\mathbf{n}}_i}{\partial \mathbf{p}_{fi}^{\mathcal{B}}} \dot{\mathbf{p}}_{fi}^{\mathcal{B}} = \frac{\partial \hat{\mathbf{n}}_i}{\partial \mathbf{p}_{fi}^{\mathcal{B}}} \lambda_i \mathbf{f}_{ti}^{\mathcal{B}} = \lambda_i \mathbf{g}_{ni}, \tag{5.18}$$

where $\mathbf{g}_{ni} = \frac{\partial \hat{\mathbf{n}}_i}{\partial \mathbf{p}_{fi}^{\mathcal{B}}} \mathbf{R}_o^T \mathbf{f}_{ti}$.

For general cases we define the stiffness matrix as $\mathbf{K}_i(\mathbf{p}_{fi}, \boldsymbol{\varsigma})$ which depends on the finger contact position and other related variables $\boldsymbol{\varsigma}$. For example $\boldsymbol{\varsigma}$ could include finger joint angles for articulated fingers or mechanical properties that determine the finger stiffness. Therefore, taking the derivative of Equation (5.1) and combining Equation (5.16) gives

$$\begin{aligned}\dot{\mathbf{f}}_{ci} &= -\dot{\mathbf{K}}_i\mathbf{d}_i - \mathbf{K}_i(\dot{\mathbf{p}}_{fi} - \dot{\mathbf{p}}_{ai}) \\ &= -\left(\frac{\partial \mathbf{K}_i}{\partial \mathbf{p}_{fi}}\dot{\mathbf{p}}_{fi} + \frac{\partial \mathbf{K}_i}{\partial \boldsymbol{\varsigma}}\dot{\boldsymbol{\varsigma}}\right)\mathbf{d}_i - \mathbf{K}_i(\mathbf{c}_{fi} + \lambda_i \mathbf{f}_{ti}) + \mathbf{K}_i\dot{\mathbf{p}}_{ai} \\ &= \lambda_i \mathbf{g}_{ci} + \mathbf{c}_{ci}, \end{aligned} \tag{5.19}$$

where $\mathbf{g}_{ci} = -\mathbf{K}_i\mathbf{f}_{ti} - \frac{\partial \mathbf{K}_i}{\partial \mathbf{p}_{fi}}\mathbf{f}_{ti}\mathbf{d}_i$, and $\mathbf{c}_{ci} = \mathbf{K}_i\dot{\mathbf{p}}_{ai} - (\frac{\partial \mathbf{K}_i}{\partial \boldsymbol{\varsigma}}\dot{\boldsymbol{\varsigma}} + \frac{\partial \mathbf{K}_i}{\partial \mathbf{p}_{fi}}\mathbf{c}_{fi})\mathbf{d}_i - \mathbf{K}_i\mathbf{c}_{fi}$. By denoting $\mathbf{h}_i = \left(\frac{\partial \mathbf{K}_i}{\partial \boldsymbol{\varsigma}}\dot{\boldsymbol{\varsigma}} + \frac{\partial \mathbf{K}_i}{\partial \mathbf{p}_{fi}}\mathbf{c}_{fi}\right)\mathbf{d}_i + \mathbf{K}_i\mathbf{c}_{fi}$, we have $\mathbf{c}_{ci} = \mathbf{K}_i\dot{\mathbf{p}}_{ai} - \mathbf{h}_i$.

Taking derivative of Equations (5.3) and (B.13) and combining Equations (5.18) and (5.19) yields

$$\begin{aligned}\dot{\mathbf{f}}_{Ni} &= \dot{\mathbf{f}}_{ci}^T\hat{\mathbf{n}}_i\hat{\mathbf{n}}_i + \mathbf{f}_{ci}^T\dot{\hat{\mathbf{n}}}_i\hat{\mathbf{n}}_i + \mathbf{f}_{ci}^T\hat{\mathbf{n}}_i\dot{\hat{\mathbf{n}}}_i \\ &= (\lambda_i\mathbf{g}_{ci} + \mathbf{c}_{ci})^T\hat{\mathbf{n}}_i\hat{\mathbf{n}}_i + \mathbf{f}_{ci}^T\lambda_i\mathbf{g}_{ni}\hat{\mathbf{n}}_i + \mathbf{f}_{ci}^T\hat{\mathbf{n}}_i\lambda_i\mathbf{g}_{ni} \\ &= \lambda_i\mathbf{g}_{Ni} + \mathbf{c}_{Ni}\end{aligned} \tag{5.20}$$

where $\mathbf{g}_{Ni} = \mathbf{g}_{ci}^T\hat{\mathbf{n}}_i\hat{\mathbf{n}}_i + \mathbf{f}_{ci}^T\mathbf{g}_{ni}\hat{\mathbf{n}}_i + \mathbf{f}_{ci}^T\hat{\mathbf{n}}_i\mathbf{g}_{ni}$, and $\mathbf{c}_{Ni} = \mathbf{c}_{ci}^T\hat{\mathbf{n}}_i\hat{\mathbf{n}}_i$. And

$$\dot{\mathbf{f}}_{ti} = \dot{\mathbf{f}}_{ci} - \dot{\mathbf{f}}_{Ni} = \lambda_i\mathbf{g}_{ti} + \mathbf{c}_{ti}, \tag{5.21}$$

where $\mathbf{g}_{ti} = \mathbf{g}_{ci} - \mathbf{g}_{Ni}$ and $\mathbf{c}_{ti} = \mathbf{c}_{ci} - \mathbf{c}_{Ni}$.

Substituting Equations (5.20) and (5.21) into (5.17) we can solve the expression for λ_i as

$$\lambda_i = \frac{\mu^2\mathbf{f}_{Ni}^T\mathbf{c}_{Ni} - \mathbf{f}_{ti}^T\mathbf{c}_{ti}}{\mathbf{f}_{ti}^T\mathbf{g}_{ti} - \mu^2\mathbf{f}_{Ni}^T\mathbf{g}_{Ni}}. \tag{5.22}$$

In the numerator, since $\mathbf{c}_{Ni} = \mathbf{c}_{ci}^T\hat{\mathbf{n}}_i\hat{\mathbf{n}}_i = (\mathbf{c}_{ci}\cdot\hat{\mathbf{n}}_i)\hat{\mathbf{n}}_i$ is along the contact normal, the term $\mathbf{f}_{Ni}^T\mathbf{c}_{Ni}$ is equivalent to $\mathbf{f}_{Ni}^T\mathbf{c}_{ci}$. By plugging in $\mathbf{c}_{ti} = \mathbf{c}_{ci} - \mathbf{c}_{Ni}$, Equation (5.22) can be simplified to

$$\begin{aligned}
\lambda_i &= \frac{\mu^2 \mathbf{f}_{Ni}^T \mathbf{c}_{ci} - \mathbf{f}_{ti}^T \mathbf{c}_{ci} + \cancelto{0\text{(orthogonal)}}{\mathbf{f}_{ti}^T \mathbf{c}_{Ni}}}{\mathbf{f}_{ti}^T \mathbf{g}_{ti} - \mu^2 \mathbf{f}_{Ni}^T \mathbf{g}_{Ni}} = \frac{\mathbf{a}_i^T \mathbf{c}_{ci}}{\lambda_{\text{den},i}} \\
&= \frac{\mathbf{a}_i^T \mathbf{K}_i}{\lambda_{\text{den},i}} \dot{\mathbf{p}}_{ai} - \frac{\mathbf{a}_i^T \mathbf{h}_i}{\lambda_{\text{den},i}} \\
&= \mathbf{g}_{\lambda_i} \dot{\mathbf{p}}_{ai} - c_{\lambda_i}, \qquad (5.23)
\end{aligned}$$

where $\mathbf{a}_i = \mu^2 \mathbf{f}_{Ni} - \mathbf{f}_{ti}$, $\lambda_{\text{den},i} = \mathbf{f}_{ti}^T \mathbf{g}_{ti} - \mu^2 \mathbf{f}_{Ni}^T \mathbf{g}_{Ni}$, $\mathbf{g}_{\lambda_i} = \mathbf{a}_i^T \mathbf{K}_i / \lambda_{\text{den},i}$ and $c_{\lambda_i} = \mathbf{a}_i^T \mathbf{h}_i / \lambda_{\text{den},i}$. The finger contact sliding velocity $\dot{\mathbf{p}}_{fi}$ can be solved for by substituting λ_i into Equation (5.16).

5.3.3. Sliding Case – Inverse Mechanics

The result of the forward mechanics, Equation (5.23), gives the finger sliding velocity for a given finger anchor velocity. For the inverse mechanics problem, we solve for the anchor motions $\dot{\mathbf{p}}_{ai}$ that cause a desired finger contact sliding velocity $\dot{\mathbf{p}}_{fi}$. Since the object motion and the contact force are given, giving a desired finger contact sliding velocity $\dot{\mathbf{p}}_{fi}$ is equivalent to giving a desired λ_i from Equation (5.16). Therefore we can write all solutions to the inverse problem as

$$\dot{\mathbf{p}}_{ai} = \mathbf{g}_{\lambda_i}^{\dagger} (\lambda_i + c_{\lambda_i}) + (\mathbf{I}^{3\times3} - \mathbf{g}_{\lambda_i}^{\dagger} \mathbf{g}_{\lambda_i}) \dot{\mathbf{p}}_{ai}^{*}, \qquad (5.24)$$

where $\mathbf{g}_{\lambda_i}^{\dagger} = \mathbf{g}_{\lambda_i}^{\mathsf{T}} (\mathbf{g}_{\lambda_i} \mathbf{g}_{\lambda_i}^{\mathsf{T}})^{-1} = \mathbf{g}_{\lambda_i}^{\mathsf{T}} / \|\mathbf{g}_{\lambda_i}\|^2$ is the pseudoinverse of $\mathbf{g}_{\lambda_i}$, $\dot{\mathbf{p}}_{ai}^{*}$ is an arbitrary anchor velocity and $(\mathbf{I} - \mathbf{g}_{\lambda_i}^{\dagger} \mathbf{g}_{\lambda_i})$ projects any vector to the nullspace of $\mathbf{g}_{\lambda_i}$.

For different solutions of $\dot{\mathbf{p}}_{ai}$, all the corresponding contact sliding velocities are the same but the changes of the contact force $\dot{\mathbf{f}}_{ci}$ are different. In Equation (5.24), $\dot{\mathbf{p}}_{ai}^{*}$ handles the redundancy of the inverse mechanics. By Equation (5.19) we can solve the corresponding contact force change $\dot{\mathbf{f}}_{ci}^{*}$ for each $\dot{\mathbf{p}}_{ai}^{*}$. The choice of $\dot{\mathbf{p}}_{ai}^{*}$ could be based on additional constraints of the anchor motions or optimizing some desired contact force properties.

5.3.4. Sliding Case Ill Condition Analysis

In the sliding mode, Equations (5.23) and (5.24) describe the relationship between the anchor motion and the contact point motion. This section analyzes specific cases where the model is ill conditioned. The key terms in the equations are $\mathbf{g}_{\lambda_i}$ and c_{λ_i}. The term $c_{\lambda_i} = \mathbf{a}_i^T \mathbf{h}_i / \lambda_{\text{den},i}$ is a scalar determined by the given system state. From Equation(5.19), when the finger stiffness is constant and the object is stationary, $\mathbf{h}_i = \mathbf{0}$ and c_{λ_i} will be zero if $\lambda_{\text{den},i}$ is nonzero.

For the other term $\mathbf{g}_{\lambda_i} = \mathbf{a}_i^T \mathbf{K}_i / \lambda_{\text{den},i}$, when the denominator $\lambda_{\text{den},i} \neq 0$, one obvious ill condition is when $\mathbf{a}_i^T \mathbf{K}_i = \mathbf{0}$. In this case the contact force is on the edge of the friction cone and the fingertip will stick on the surface no matter how the anchor moves.

Proposition 1. *The product of* $\mathbf{a}_i^T \mathbf{K}_i$ *will be nonzero when the stiffness matrix* $\mathbf{K}_i$ *is positive definite.*

Proof. With assumption 4) the term $\mathbf{a}_i = \mu^2 \mathbf{f}_{Ni} - \mathbf{f}_{ti}$ will be nonzero, therefore Proposition 1 holds since the matrix $\mathbf{K}_i$ is full rank when it is positive definite. □

For the denominator term $\lambda_{\text{den},i} = \mathbf{f}_{ti}^T \mathbf{g}_{ti} - \mu^2 \mathbf{f}_{Ni}^T \mathbf{g}_{Ni}$, besides given current positions, velocities and corresponding contact force, it is also determined by the stiffness $\mathbf{K}_i$, stiffness variances $\frac{\partial \mathbf{K}_i}{\partial \mathbf{p}_{fi}}$ and $\frac{\partial \mathbf{K}_i}{\partial \boldsymbol{\varsigma}}$ and change of the normal direction $\frac{\partial \hat{\mathbf{n}}_i}{\partial \mathbf{p}_{fi}^{\mathcal{B}}}$. When the denominator $\lambda_{\text{den},i} = 0$, the sliding velocity will go to infinity and the quasistatic assumption will be violated as we have shown in the unstable sliding example of Section 5.2. Now we focus on a case where finger stiffness and contact normal are fixed and derive conditions that avoid unstable sliding.

Theorem 2. *When the finger stiffness matrix and contact normal direction are constant, the denominator term $\lambda_{den,i}$ will be nonzero when the stiffness matrix $\mathbf{K}_i$ is symmetric positive definite.*

Proof. When $\frac{\partial \mathbf{K}_i}{\partial \mathbf{p}_{fi}} = 0$, $\frac{\partial \mathbf{K}_i}{\partial \varsigma} = 0$ and $\frac{\partial \hat{\mathbf{n}}_i}{\partial \mathbf{p}_{fi}^{\mathcal{B}}} = 0$, the key variables in Equations (5.20) and (5.21) will become

$$\mathbf{g}_{ci} = -\mathbf{K}_i \mathbf{f}_{ti} \text{ and } \mathbf{g}_{Ni} = -\mathbf{f}_{ti}^T \mathbf{K}_i^T \hat{\mathbf{n}}_i \hat{\mathbf{n}}_i. \tag{5.25}$$

Because $\mathbf{g}_{Ni}$ and $\mathbf{f}_{Ni}$ are both vectors in the direction of $\hat{\mathbf{n}}_i$, we have $\mathbf{f}_{Ni}^T \mathbf{g}_{Ni} = -\|\mathbf{f}_{Ni}\| \mathbf{f}_{ti}^T \mathbf{K}_i^T \hat{\mathbf{n}}_i$. Plugging Equation (5.25) into (5.23), we have

$$\begin{aligned} \lambda_{\text{den},i} &= \mathbf{f}_{ti}^T (\mathbf{g}_{ci} - \mathbf{g}_{Ni}) - \mu^2 \mathbf{f}_{Ni}^T \mathbf{g}_{Ni} \\ &= -\mathbf{f}_{ti}^T \mathbf{K}_i \mathbf{f}_{ti} + \mu^2 \mathbf{f}_{ti}^T \mathbf{K}_i^T \mathbf{f}_{Ni}. \end{aligned}$$

When $\mathbf{K}_i$ is symmetric,

$$\lambda_{\text{den},i} = \mathbf{f}_{ti}^T \mathbf{K}_i (\mu^2 \mathbf{f}_{Ni} - \hat{\mathbf{f}}_{ti}) = \mathbf{f}_{ti}^T \mathbf{K}_i \mathbf{a}_i, \tag{5.26}$$

where $\mathbf{f}_{ti}$ and $\mathbf{a}_i$ are both nonzero due to assumption 4. Therefore similar to the proof of Proposition 1, when $\mathbf{K}_i$ is positive definite, $\lambda_{\text{den},i}$ will be nonzero. □

Note that the ill conditions discussed in Proposition 1 and Theorem 2 are excluded by the assumption of symmetric positive definite stiffness in Section 5.1.1.

5.4. Object Mechanics

In the previous section, we study the contact mechanics of a single finger with respect to given object positions and velocities. In this section we extend the previous work to consider all the finger contacts and the external contacts with the environment. For given the motions of the object and all the anchors, we solve for object velocity constraints that maintain external contacts and force-balance constraint in terms of the forces from the fingers and the external contacts.

5.4.1. Object Velocity Constraints

For the given external contact locations, the object motion need to satisfy that all the external contacts remain fixed to the object, i.e., no motions that break any external contact or penetrate into the environment is allowed.

We denote the contact point position of the jth ($j = 1, ..., m$) external contact as $\mathbf{p}_{ej}$ in $\mathcal{W}$, and $\mathbf{p}_{ej}^{\mathcal{B}}$ in $\mathcal{B}$. we can write the jth contact point velocity as

$$\dot{\mathbf{p}}_{ej} = \dot{\mathbf{p}}_o + [\boldsymbol{\omega}_o]\mathbf{R}_o\mathbf{p}_{ej}^{\mathcal{B}} + \mathbf{R}_o\dot{\mathbf{p}}_{ej}^{\mathcal{B}}. \tag{5.27}$$

Since we assume the contact point cannot move in the contact normal direction, we have

$$\dot{\mathbf{p}}_{ej}^T\hat{\mathbf{n}}_j = 0, \tag{5.28}$$

where $\hat{\mathbf{n}}_j$ is the normal direction at the jth external contact. For all the external contacts, there are m object velocity constraints in total.

5.4.2. Object Force Constraint

Considering the contacts from both the fingers and the external environment, the object force constraint refers to that all the wrenches applied to the object should be balanced at all times in order to maintain the quasistatic assumption. For the jth external contact, the contact force applied to the object is denoted as $\mathbf{f}_{ej}$. When the contact point is sticking, its friction cone is approximated with n_c-sided right pyramids where $n_c \geq 3$ for the spatial case and $n_c = 2$ for the planar case. The corresponding external contact wrench is $\mathbf{w}_{ej} = [(\mathbf{p}_{ej} \times \mathbf{f}_{ej})^T, \mathbf{f}_{ej}^T]^T$. The friction cones are turned into a wrench cone of the object, denoted by $\mathcal{WC}_e$. The wrench $\mathbf{w}_{ej}$ can be expressed as the conical combination of basis vectors:

$$\mathbf{w}_{ej} = \sum_{k=1}^{n_c} \beta_{jk} \hat{\mathbf{w}}_{jk}, \;\; \beta_{jk} \geq 0, \tag{5.29}$$

where $\hat{\mathbf{w}}_{jk}, k = 1, ..., n_c$ denote basis vectors of $\mathcal{WC}_e$. We denote $\mathbf{w}_e$ as the sum of all the external contact wrenches:

$$\mathbf{w}_e = \sum_{j=1}^{m} \mathbf{w}_{ej} = \hat{\mathbf{W}} \boldsymbol{\beta}, \tag{5.30}$$

where $\hat{\mathbf{W}} = [\hat{\mathbf{w}}_{11}, \hat{\mathbf{w}}_{12}, \ldots, \hat{\mathbf{w}}_{mn_c},] \in \mathbb{R}^{6 \times mn_c}$ and $\boldsymbol{\beta} = [\beta_{11}, \beta_{12}, \ldots, \beta_{mn_c}]^T \in \mathbb{R}^{mn_c \times 1}$.

For the finger contact force $\mathbf{f}_{ci}$, the corresponding wrench applied to the object is

$$\mathbf{w}_{ci} = [(\mathbf{p}_{fi} \times \mathbf{f}_{ci})^T, \mathbf{f}_{ci}^T]^T. \tag{5.31}$$

Therefore the object force balance condition can be written as

$$\mathbf{w}_c + \mathbf{w}_e + \mathbf{w}_g = \mathbf{0}, \tag{5.32}$$

where $\mathbf{w}_c = \sum_{i=1}^{n} \mathbf{w}_{ci}$ is the total finger contact wrench and $\mathbf{w}_g$ is the gravitational wrench. Equation (5.32) describes the object force constraint related to the object and finger anchor motions.

A set of object and anchor trajectories are feasible when both the object velocity (5.28) and force constraints (5.32) are satisfied.

5.5. Robustness Analysis

System performance can be affected by state errors due to imperfect trajectory tracking and uncertainty in estimated parameters such as friction coefficients and object geometry. In this section we discuss conditions that ensure that a given set of object and anchor trajectories will be robust to these uncertainties. Due to the spring compliance, from Equations (5.1) and (5.31) uncertainties in anchor motions, finger stiffnesses and finger contact point positions would result in a contact wrench uncertainty $\delta\mathbf{w}_c$.

Definition 1. *A set of object and anchor trajectories is robust to a contact wrench uncertainty $\delta\mathbf{w}_c$, when there always exists a external contact wrench $\delta\mathbf{w}_e = -\delta\mathbf{w}_c$ that can guarantee the force balance of the object (Equation (5.32)).*

The external contact wrenches $\mathbf{w}_e$ are bounded by the external frictional wrench cone $\mathcal{WC}_e$. To guarantee a resistant $\delta\mathbf{w}_e$ exist, $\mathbf{w}_e + \delta\mathbf{w}_e$ should also lie in $\mathcal{WC}_e$. Therefore the further away $\mathbf{w}_e$ is from the edges of $\mathcal{WC}_e$ the more robust the trajectory will be.

Theorem 3. *Consider a set of object and anchor trajectories with resulting contact wrenches $\mathbf{w}_c(t) = \bar{\mathbf{w}}_c(t) + \delta\mathbf{w}_c$, where $\bar{\mathbf{w}}_c(t)$ are the nominal finger contact wrench trajectories and $\delta\mathbf{w}_c$ are random wrench uncertainties with $-\varepsilon\mathbf{1} \leq \delta\mathbf{w}_c \leq \varepsilon\mathbf{1}$, where $\varepsilon > 0$ and $\mathbf{1}^{6\times 1}$ is column vector with all ones. The trajectories are robust to the bounded wrench uncertainty when $\hat{\mathbf{W}}$ is full rank and $\boldsymbol{\beta} \geq \varepsilon\|\hat{\mathbf{W}}^{\dagger}\mathbf{1}\|$ at all times, where $\hat{\mathbf{W}}^{\dagger} = \hat{\mathbf{W}}^T(\hat{\mathbf{W}}\hat{\mathbf{W}}^T)^{-1}$.*

Proof. Since the uncertainty $\delta\mathbf{w}_c$ spans all dimensions of the wrench space, when $\hat{\mathbf{W}}$ is not full rank it is impossible to guarantee the counter wrench $-\delta\mathbf{w}_c$ can always be provided by the external contacts.

From Definition 1 and Equation (5.30), the resistant external contact wrench is

$$\begin{aligned} \delta\mathbf{w}_e &= \hat{\mathbf{W}}\delta\boldsymbol{\beta} = -\delta\mathbf{w}_c \\ &\rightarrow \ \delta\boldsymbol{\beta} = -\hat{\mathbf{W}}^{\dagger}\delta\mathbf{w}_c. \end{aligned} \tag{5.33}$$

To satisfy the Coulomb friction assumption we have

$$\boldsymbol{\beta} + \delta\boldsymbol{\beta} \geq \mathbf{0}. \tag{5.34}$$

Substituting Equation (5.33) to (5.34) gives

$$\boldsymbol{\beta} \geq \hat{\mathbf{W}}^{\dagger}\delta\mathbf{w}_c. \tag{5.35}$$

Since $-\varepsilon\mathbf{1} \leq \delta\mathbf{w}_c \leq \varepsilon\mathbf{1}$, if Equation (5.35) holds for any $\delta\mathbf{w}_c$, $\boldsymbol{\beta}$ should satisfy

$$\boldsymbol{\beta} \geq \varepsilon\|\hat{\mathbf{W}}^{\dagger}\mathbf{1}\|. \tag{5.36}$$

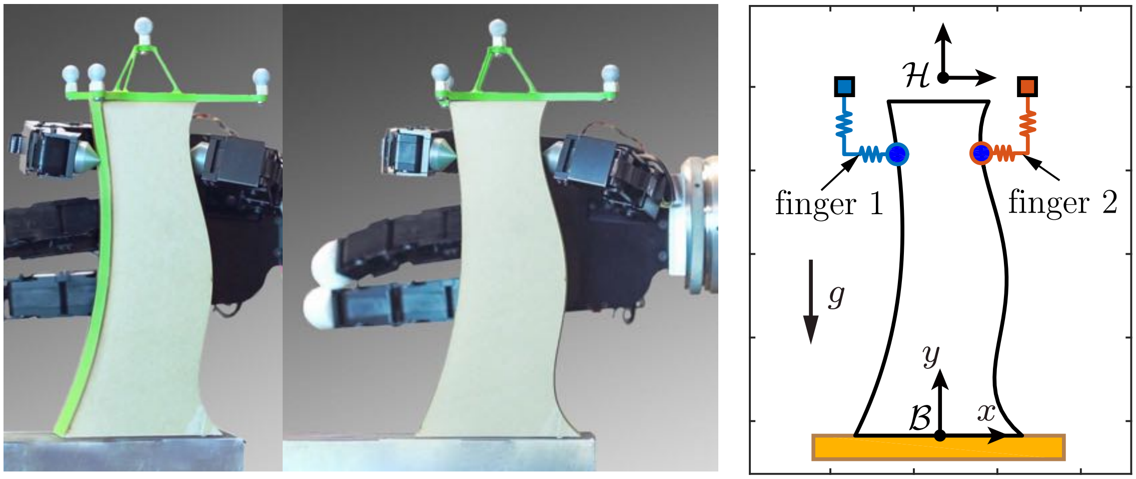
□

5.6. Motion Planning

In previous sections, we provide tools to model the system of an n-fingered hand regrasping an object and analyze the robustness of given trajectories. In this section, we discuss the motion planning problem used to generate candidate trajectories. We focus on the cases where the object external contacts are given and remain in a single contact mode. The goal is to find feasible and robust object and finger anchor motions that can achieve the desired regrasp. We propose a general motion planning framework as follows.

For initial grasps which are force-balanced and with all the fingers sticking, we define regrasp motions that consist of two phases: a *sticking phase* from time $0 \leq t_1 < T_1$ and a *sliding phase* from time $0 \leq t_2 \leq T_2$. In the sticking phase we plan the anchor and object motions that can move the finger contact forces to the edges of the friction cones. At the end of the sticking phase, the tangential contact forces applied to the object should along the initial sliding directions. In the sliding phase, we choose desired finger contact and object trajectories such that the fingertips slide to the goal. Using the result of Section 5.3, along with the initial sliding condition given from the last state of the sticking phase, we can calculate the anchor trajectories.

During both the sticking and sliding phases, the object and sliding motions should be chosen such that the trajectories are feasible and robust as discussed in Sections 5.4



(a) hardware setup (b) configuration (0.05 m/div)

Figure 5.8. The planar system: (a) shows the Allegro hand grasping an extruded object sitting on a fixed table, (b) shows the configuration of the system. The positions of the virtual anchors (squares) are fixed in the hand frame $\mathcal{H}$. The colored lines between the anchors and contact points (circles) show the programmed springs.

and 5.5. Additional constraints and optimal cost-to-go could be introduced to handle the motion redundancy when planning the anchor motions. In the next section, we detail a planar system to show an implementation of the proposed algorithm.

5.7. Implementation

In this section we detail a two-fingered planar system and specify the motion planning algorithm in the previous chapter to find anchor trajectories that can realize a desired regrasp and maximize the robustness to wrench uncertainty. We assume the object remains stationary. We identify modeling parameters from experimental data using the forward contact mechanics model. We implement the motion planning algorithm with the identified parameters and comparisons between simulated sliding trajectories and experimental result are shown.

5.7.1. Description of A Planar System

The system consists of a two-fingered hand grasping an object with smooth edges (Figure 5.8). The object sits on a fixed table, and the motions of the hand are in the vertical plane. Let $\mathcal{H}$ denote a hand frame attached to the hand and originating at $\mathbf{p}_h$. The finger stiffnesses and anchor positions are assumed to be fixed in $\mathcal{H}$, i.e., $\mathbf{K}_i^{\mathcal{H}}$ and $\mathbf{p}_{ai}^{\mathcal{H}}$ are constant. Therefore the two anchor positions are uniquely determined by the hand positions and in-hand sliding is realized by controlling the hand motion. For simplicity, we allow only translational hand motions and assume the object is stationary. By adding these constraints the inverse mechanics problem in Section 5.3.3 will yield a unique solution of anchor motions.

The experiments are implemented with the ERIN system as introduced in Chapter 3. we use two fingers of the Allegro hand to grasp the extruded object. Each finger consists of four joints which are individually controlled by geared DC motors. The fingers are joint-torque controlled at 333 Hz to perform desired fingertip springiness $\mathbf{K}_i^{\mathcal{H}}$. Deviations from fingertip positions $\mathbf{p}_{fi}^{\mathcal{H}}$ to their programmed equilibrium points $\mathbf{p}_{0i}^{\mathcal{H}}$ multiplied by $\mathbf{K}_i^{\mathcal{H}}$ give desired fingertip forces. The forces are turned into referenced joint torques by multiplying transpose of Jacobian matrix:

$$\boldsymbol{\tau}_i = \mathbf{J}_i^T \left[\mathbf{K}_i^{\mathcal{H}} (\mathbf{p}_{0i}^{\mathcal{H}} - \mathbf{p}_{fi}^{\mathcal{H}}) \right], \tag{5.37}$$

where $\boldsymbol{\tau}_i$ denotes referenced joint torques for the ith finger and $\mathbf{J}_i$ denotes the Jacobian matrix. Fingertip positions $\mathbf{p}_{fi}^{\mathcal{H}}$ and $\mathbf{J}_i$ are evaluated with joint encoder feedback. The hand is mounted at the end of the WAM arm which is controlled at 500 Hz. Positions of the hand and the object are both sensed by the motion capture system.

This method of simulating contact compliance using torque control of the finger joints has a number of tradeoffs with using a passively compliant hand. It has advantages of

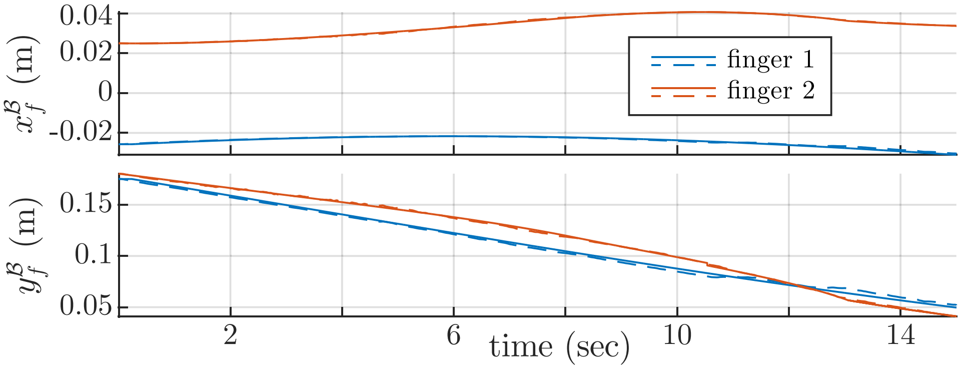


Figure 5.9. Parameter fitting result of finger contact point position trajectories $\mathbf{p}_{fi}^{\mathcal{B}}(t)$. Dashed lines are experimental data and solid lines are fitting results.

programmability and compatibility with other dexterous robotic hands, but may introduce errors due to control bandwidth limitations, joint friction and backlash, and low encoder resolution.

5.7.2. Parameter Identification

By using knowledge of the forward contact mechanics from Section 5.3.2, given anchor and object motions we can simulate the fingertip motions from initial states. Therefore with this model we can identify modeling parameters using experimental data. Besides the finger contact friction coefficient μ, we also included the finger stiffness parameters ($\mathbf{K}_i^{\mathcal{H}}$ and $\mathbf{p}_{0i}^{\mathcal{H}}$) to be identified due to the uncertainties from the Allegro hand.

The initial grasp was manually positioned and the hand was commanded to move in the $-y$-direction for 0.15 m. Optimal parameters were fitted by a SQP solver and the cost function was chosen to be the sum of absolute errors between simulated and experiment finger contact positions. We used data from a 10 s long experiment with 5000 samples of fingertip positions. Result of the parameter identification can be found in Table 5.1. And

parameters	initial guess	optimal
μ	0.24	0.2502
$\mathbf{K}_1^{\mathcal{H}}$ (N/m)	$\begin{bmatrix} 150 & 0 \\ 0 & 100 \end{bmatrix}$	$\begin{bmatrix} 152.06 & 0 \\ 0 & 101.1 \end{bmatrix}$
$\mathbf{K}_2^{\mathcal{H}}$ (N/m)	$\begin{bmatrix} 150 & 0 \\ 0 & 100 \end{bmatrix}$	$\begin{bmatrix} 150.23 & 0 \\ 0 & 105.94 \end{bmatrix}$
$\mathbf{p}_{01}^{\mathcal{H}}$ (m)	$[-0.025,\ 0.04]^T$	$[-0.0255,\ 0.0397]^T$
$\mathbf{p}_{02}^{\mathcal{H}}$ (m)	$[0,\ 0.04]^T$	$[0.0006,\ 0.0412]^T$

Table 5.1. Parameter identification result.

Figure 5.9 shows the finger contact position trajectories from both experiment and fitting results.

5.7.3. Planar Robust Regrasp

In this section we show the planning and execution of a planar regrasp task using the parameters identified above. The initial grasp is manually positioned and measured by the motion capture system. Based on Section 5.6, we specify a motion planning algorithm to find feasible hand motions to realize the desired regrasp and maximize robustness. The planned hand trajectories are executed by the WAM arm.

For the planar system, when both the fingers are sliding, we know the contact forces directions if given the contact positions and sliding directions. Due to redundancy in the inverse mechanics, there are two remaining freedoms that control the magnitude of the contact forces. Since we assume the anchors are fixed to the hand and the hand is only allowed translational motions, there will be a unique solution for hand position $\mathbf{p}_h$

in terms of given contact points and sliding directions. The expression for $\mathbf{p}_h$ is derived below.

The mapping of anchor positions from $\mathcal{H}$ to $\mathcal{W}$ is

$$\mathbf{p}_{ai} = \mathbf{p}_h + \mathbf{R}_h \mathbf{p}_{ai}^{\mathcal{H}}, \tag{5.38}$$

where $\mathbf{R}_h$ is the rotation matrix of $\mathcal{H}$. Based on the previous assumptions, $\mathbf{R}_h$ and $\mathbf{p}_{ai}^{\mathcal{H}}$ are fixed. We denote ${}^{\perp}\hat{\mathbf{f}}_{ci}$ as the perpendicular direction of the contact force $\mathbf{f}_{ci}$, and we have

$${}^{\perp}\hat{\mathbf{f}}_{ci} \cdot \mathbf{f}_{ci} = 0 \rightarrow {}^{\perp}\hat{\mathbf{f}}_{ci}^{T} \mathbf{f}_{ci} = 0. \tag{5.39}$$

Since $\mathbf{f}_{ci}$ is along the edge of the friction cone, given a finger contact position, ${}^{\perp}\hat{\mathbf{f}}_{ci}$ can be obtained from the object geometry, contact friction, and sliding direction information. Substituting Equations (5.1) and (5.38) to (5.39), we can solve the hand position for a given pair of finger contact positions $\{\mathbf{p}_{f1}, \mathbf{p}_{f2}\}$ as

$$\mathbf{p}_h = \begin{bmatrix} {}^{\perp}\hat{\mathbf{f}}_{c1}^{T} \mathbf{K}_1 \\ {}^{\perp}\hat{\mathbf{f}}_{c2}^{T} \mathbf{K}_2 \end{bmatrix}^{-1} \begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix}, \tag{5.40}$$

where $\Delta_i = {}^{\perp}\hat{\mathbf{f}}_{ci}^{T} \mathbf{K}_i (\mathbf{p}_{fi} - \mathbf{R}_h \mathbf{p}_{ai}^{\mathcal{H}} - \mathbf{d}_{0i})$.

With the solved $\mathbf{p}_h$, corresponding fingertip contact forces can be solved for using Equation (5.1). Combined with Equations (5.31) and (5.32), we can also test if the corresponding fingertip contact wrenches can be balanced by the external contacts.

5.7.3.1. Finger Contact Position Map. For the given object, the finger contact positions can be uniquely determined by the y-positions of the fingertips in the object frame $\mathcal{B}$. Given a pair of $\{y_{f1}^{\mathcal{B}}, y_{f2}^{\mathcal{B}}\}$, we can calculate the contact positions $\{\mathbf{p}_{f1}^{\mathcal{B}}, \mathbf{p}_{f2}^{\mathcal{B}}\}$, and given the sliding directions we can determine the corresponding anchor and hand positions from

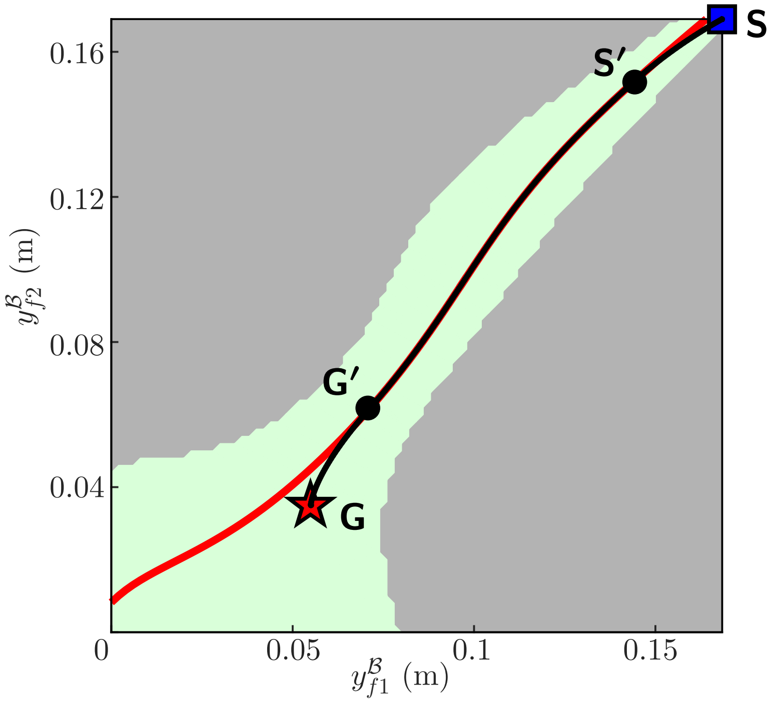


Figure 5.10. A diagram of a robust task space sliding regrasp plan. The diagram shows a finger contact position map (FCmap) where the green shaded part is the feasible region where the object is in force balance. The two fingers both moves in the $-y$-direction of $\mathcal{B}$. The red curve shows the most robust curve $\boldsymbol{\xi}^*$. The black curve is the planned path for the regrasp.

Equation (5.40). Therefore the full task space of the planning problem can be represented by the $y_{fi}^{\mathcal{B}}$ positions. Figure 5.10 shows the finger contact position map (FCmap): the start point is at $\mathbf{S} = [0.1684\,\text{m},\ 0.169\,\text{m}]^T$ and the goal is to reach $\mathbf{G} = [0.055\,\text{m},\ 0.035\,\text{m}]^T$. The sliding directions determine which friction cone edges the contact forces lie on. In this example we assume the fingers always slide toward the goal. For each point on the map, we can get the contact position $\mathbf{p}_{fi}$ from object position and geometry, then calculate hand position $\mathbf{p}_h$, anchor positions $\mathbf{p}_{ai}$ and contact forces $\mathbf{f}_{ci}$ with Equations (5.40) (5.38) and (5.1). Finally based on Equations (5.31) and (5.32), we can test if the contact forces could be balanced by the external contacts with a linear program:

$$\min_{\boldsymbol{\beta}} \mathbf{1}^T\boldsymbol{\beta}, \text{ subject to} \begin{cases} \hat{\mathbf{W}}\boldsymbol{\beta} = -\mathbf{w}_c - \mathbf{w}_g \\ \boldsymbol{\beta} \geq \mathbf{0}^{mn_c\times 1} \end{cases}, \tag{5.41}$$

where in this example $m = 2$ and $n_c = 2$, and $\mathbf{1}$ is a 4×1 column vector with all ones. If a solution of $\boldsymbol{\beta}$ is found, the contact positions satisfy the force balance constraint. In Figure 5.10, feasible contact point positions are marked as green. The friction coefficient between the object and the table is $\mu_e = 1$ and the gravity of the object is $-10.1\,\mathrm{N}$.

5.7.3.2. Planning Algorithm. In this example the desired regrasps are realized by planning the hand motion $\mathbf{p}_h(t)$. Initial grasps are assumed to be force-balanced and with all the fingers sticking. The first step is to test if the goal position G is in the feasible region with (5.41). If the goal is out of the feasible region, the desired regrasp cannot be achieved with the current external contacts. Following Section 5.6, the planning algorithm is detailed below.

Sticking Phase: we define the hand trajectories $\mathbf{p}_h(t_1)$ as cubic time polynomials and set velocity boundary conditions $\dot{\mathbf{p}}_h(0) = \dot{\mathbf{p}}_h(T_1) = \mathbf{0}$. Given initial condition S and goal G, the hand position at the end of the sticking phase $\mathbf{p}_h(T_1)$ can be calculated with Equation (5.40). The trajectories of the sticking phase are determined as long as the period T_1 is given.

Sliding Phase: Since the fingertip contact positions can be described by the coordinates $\{y^{\mathcal{B}}_{f1}, y^{\mathcal{B}}_{f2}\}$, we use $\boldsymbol{\xi}(t_2) = [y^{\mathcal{B}}_{f1}(t_2),\ y^{\mathcal{B}}_{f2}(t_2)]^T$ to represent sliding trajectories. To accomplish the desired regrasp we have $\boldsymbol{\xi}(0) = \mathsf{S}$ and $\boldsymbol{\xi}(T_2) = \mathsf{G}$. From the robust analysis in Section 5.5, sliding trajectories are feasible as long as $\boldsymbol{\xi}(t_2)$ always lies in the feasible region of FCmap. Based on the findings in Section 5.5, the further away the required external contact wrench $\mathbf{w}_e$ is from the boundaries of $\mathcal{WC}_e$, the more robust the current fingertip relative positions are. In this system given the environment contacts and the

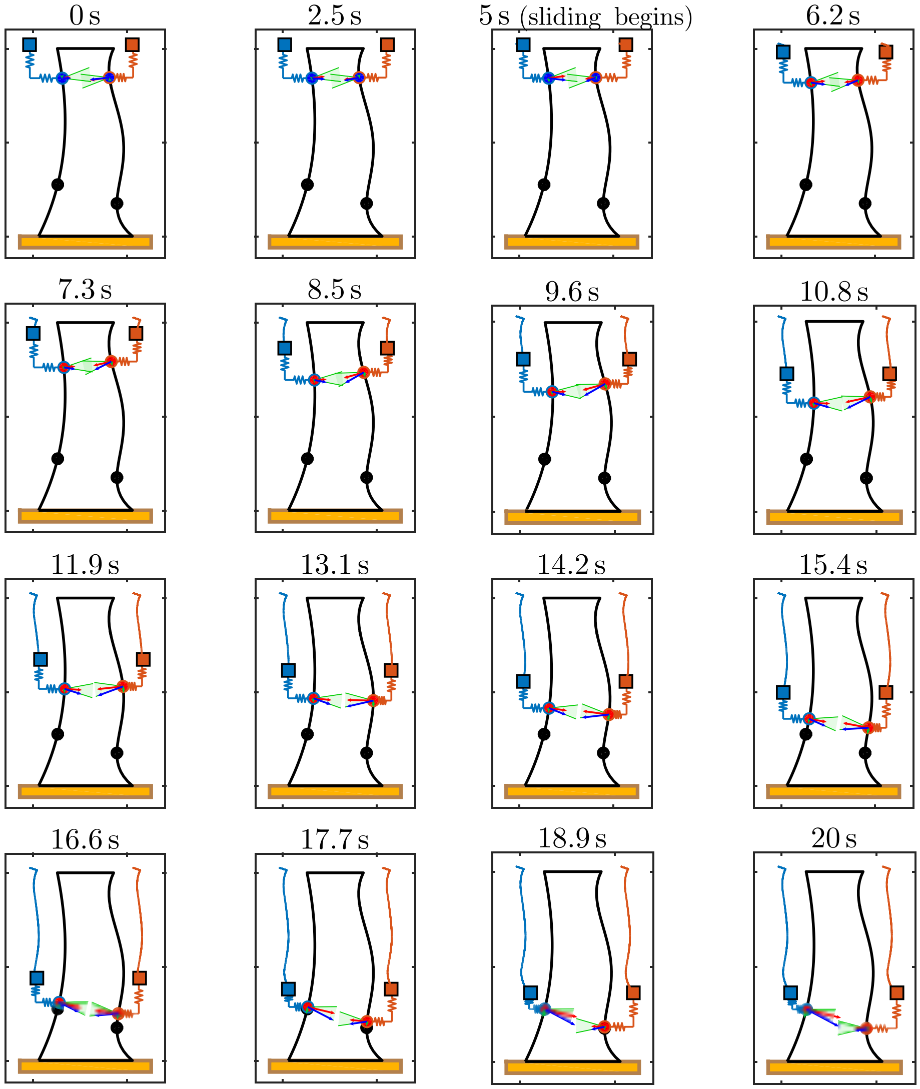


Figure 5.11. Snapshots of the planned motions. Positions are in meters (0.1 m/div). Small squares show the finger anchors. Blue (sticking) and red (sliding) dots show finger contact points from simulation. Black dots are the goal contact positions. Green lines shows the edges of the contact friction cones. Blue, red and green arrows show contact forces, contact normal forces and contact tangential forces respectively.

contact position of one finger, there is an optimally robust contact position of the other finger. The union of the most robust fingertip position pairs draws a curve in FCmap,

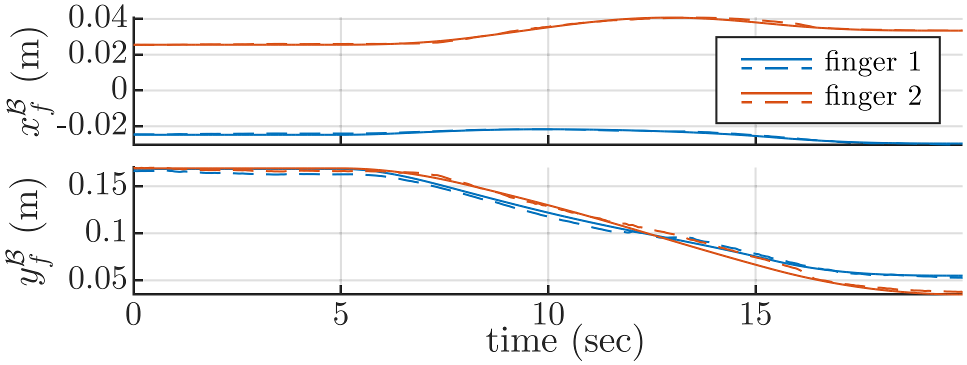


Figure 5.12. Experimental result showing contact point positions in $\mathcal{B}$. Dashed lines are experimental data and solid lines are simulated trajectories.

denoted by $\boldsymbol{\xi}^*$. To describe how far a wrench $\mathbf{w}_e$ is to the faces of the wrench cone $\mathcal{WC}_e$, we define a matrix ${}^{\perp}\hat{\mathbf{W}}$ whose rows are unit vectors normal to the faces of $\mathcal{WC}_e$ and pointing into the cone. The path of $\boldsymbol{\xi}^*$ is then found by the following procedure:

$$\forall\, y_{f1}^{\mathcal{B}},\ \text{find } y_{f2}^{\mathcal{B}*} \text{ such that}$$

$$\max d,\ \text{subject to} \begin{cases} {}^{\perp}\hat{\mathbf{W}}\hat{\mathbf{w}}_e \geq d \\ d \geq 0 \end{cases},$$

where $\hat{\mathbf{w}}_e$ are the normalized total external contact wrenches. The solved $\boldsymbol{\xi}^*$ is shown as the red curve in Figure 5.10.

To maximize robustness, the principle of our planning algorithm is to plan a $\boldsymbol{\xi}$ that overlaps with $\boldsymbol{\xi}^*$ as much as possible. By introducing a point $\mathbf{S}'$ where $\boldsymbol{\xi}$ reaches $\boldsymbol{\xi}^*$ and a point $\mathbf{G}'$ where $\boldsymbol{\xi}$ leaves $\boldsymbol{\xi}^*$, the sliding trajectory $\boldsymbol{\xi}(t_2)$ is defined by three pieces:

- *1st piece* $(\mathbf{S} \to \mathbf{S}',\ 0 \leq t_2 \leq T_{21})$: The contact sliding trajectories $\boldsymbol{\xi}(t_2)$ are cubic time polynomials. The boundary conditions are $\boldsymbol{\xi}(0) = \mathbf{S}$, $\boldsymbol{\xi}(T_{21}) = \mathbf{S}'$, $\dot{\boldsymbol{\xi}}(0) = \mathbf{0}$, and $\dot{\boldsymbol{\xi}}(T_{21}) = \mathbf{v}_s$, where $\mathbf{v}_s$ is determined by the initial velocity of the next piece.

- *2nd piece* $(\mathbf{S}' \to \mathbf{G}',\ T_{21} \le t_2 \le T_{21} + T_{22})$: The contacts slide along $\boldsymbol{\xi}^*$ for a period of T_{22}. The sliding velocities are assumed to have a constant magnitude $\|\dot{\boldsymbol{\xi}}\| = v_2 = L_2/T_{22}$ where L_2 is the arc length of $\boldsymbol{\xi}^*$ between $\mathbf{S}'$ and $\mathbf{G}'$. The initial and final velocities are $\mathbf{v}_s = v_2\, \partial\hat{\boldsymbol{\xi}}^*|_{\mathbf{S}'}$ and $\mathbf{v}_g = v_2\, \partial\hat{\boldsymbol{\xi}}^*|_{\mathbf{G}'}$, where $\partial\hat{\boldsymbol{\xi}}^*|_{\mathbf{X}}$ means the normalized tangent vector at point $\mathbf{X}$.
- *3rd piece* $(\mathbf{G}' \to \mathbf{G},\ T_{21} + T_{22} \le t_2 \le T_2)$: The contacts slide from $\mathbf{G}'$ to $\mathbf{G}$ following cubic time polynomials. The boundary conditions are $\boldsymbol{\xi}(T_{21}+T_{22}) = \mathbf{G}'$, $\boldsymbol{\xi}(T_2) = \mathbf{G}$, $\dot{\boldsymbol{\xi}}(T_{21} + T_{22}) = \mathbf{v}_g$, and $\dot{\boldsymbol{\xi}}(T_2) = \mathbf{0}$.

With the definitions above, the motion planning problem can be described by the following nonlinear program:

$$\begin{aligned} &\max_{\mathbf{S}',\mathbf{G}',T_{21},T_{22}} \; L_2(\boldsymbol{\xi}^*, \mathbf{S}', \mathbf{G}') - \kappa V_{\max}(\mathbf{S}, \mathbf{G}, \mathbf{S}', \mathbf{G}', T_{21}, T_{22}), \\ &\text{subject to} \begin{cases} \mathtt{sgn}(\dot{\boldsymbol{\xi}}) = \mathtt{sgn}(\mathbf{G} - \mathbf{S}) \\ T_{21} + T_{22} \le T_2, \end{cases}, \end{aligned} \tag{5.42}$$

where κ is a positive weighting scalar and $V_{\max} = [1,\ 1]\ \mathtt{max}(\dot{\boldsymbol{\xi}})$ represents the maximum sliding velocities. The first constraint ensures that the sliding directions are always towards the goal as we assumed in 5.7.3.1.

Given $T_1 = 5\,\mathrm{s}$, $T_2 = 15\,\mathrm{s}$ and $\kappa = 0.5$, the planned contact sliding path is shown in Figure 5.10, and corresponding trajectories of the fingers are shown in Figure 5.11.

The planned trajectories were executed and the results are shown in Figure 5.12. The final relative fingertip position errors are $[2.2\,\mathrm{mm}, 2.6\,\mathrm{mm}]$ w.r.t. total travel distances of $[114.2\,\mathrm{mm}, 136.3\,\mathrm{mm}]$ for the two fingers respectively.

CHAPTER 6

Conclusion

In this thesis we study in-hand manipulation by controlled sliding to reposition finger contact positions relative to a grasped object. From different sources of external forces and contact types, we investigate two versions of the problem.

In Chapter 4, we presented a general framework for planning dynamic in-hand sliding manipulation motions and analyzed the dynamics for n-fingered grasps using soft-finger limit surface models. We proposed a simple model of the contact pressure distribution and constructed the frictional limit surfaces based on it. The framework was applied to the problem of in-hand sliding manipulation with a two-fingered grasp in the horizontal plane. Our motion planner was able to automatically find dynamic hand motions to achieve a desired sliding regrasp based on the grasp contact model. Experimental implementations of iterative planning and execution reduced the relative position error and demonstrated the feasibility of the overall approach.

In Chapter 5, we showed that by combining spring and sliding compliances, regrasping by in-hand sliding with external contacts can be realized without haptic feedback. The finger spring compliance ensures contact while sliding, and allows contact force control by only controlling the finger anchor motions. The proposed approach is validated with a planar example and the experiment result shows less than 2% residual errors following the execution of planned trajectories.

6.1. Future Directions

In this thesis, both methods for performing in-hand manipulation were based on system modeling, motion planning, and then experimental validation. In Chapter 4 we use iterative replanning to reduce the residual errors and in Chapter 5 the open loop plans were shown to yield small errors. A future research direction for both methods is to further reduce errors, and this can be done in numerous ways.Firstly, more advanced modeling and parameter estimation could help. For example, the accuracy of the Coulomb friction model is limited to specific cases. A good model should balance adaptability, accuracy and complexity. As addressed in topics 5) and 6) in Chapter 4, another method of error reduction during dynamic sliding regrasps is to add feedback control to actively correct planned motions during execution. This will likely require better fingertip force sensors and higher-bandwidth fingertip force control than is currently available to us. The performance of the feedback during dynamic regrasps is also limited by contact model uncertainties, trajectory tracking performance of the robot, and object state estimation. For example, Theorem 1, which promises zero ultimate error for an idealized 1-DOF regrasping problem, is based on perfect trajectory tracking, perfect sensing, and a constant (though unknown) friction coefficient. Each of these assumptions should be weakened to better understand the importance of each factor.

Another future research direction is to study error-corrective sliding for assembly tasks. The problem is to choose a grasp configuration that satisfies force-closure constraints and provides error-corrective sliding motion in response to likely disturbance forces during the place operation. For example, uncertainty in a peg-in-hole assembly task results in contact forces that should be mapped to error-corrective motion, using the remote center of compliance (RCC) device [**65**] or using active accommodation control [**30**]. Alternatively, it is possible to use sliding at the fingertips as the source of compliance. By the choice of finger locations and normal forces, we can control the shape of the grasp limit surface (Section 4.3.2), which governs the mapping from contact forces to sliding directions, much like an accommodation control law maps contact forces to corrective velocities.

For Chapter 5 future work could modify the point contact model to allow regrasps for more general patch contact grippers. For other compliant hands, the difficulty of estimating the stiffness parameters $\mathbf{K}_i(\mathbf{p}_{fi}, \boldsymbol{\varsigma})$ may vary dramatically depending on the specific structure of the hands. The forward contact mechanics analysis of Section 5.7.2 can be generalized to other grippers so that the finger stiffnesses can be identified with experimental data.

For the example in Section 5.7, we prespecified the external contact locations and finger contact mode sequences. In future work the motion planning algorithm could be expanded to judiciously choose the external contacts and sequences of fingertip sticking and sliding phases to add more design freedoms for the anchor motions in order to optimize other desired properties. Future works should focus on describing the reachable set of each contact mode and deriving efficient planning algorithms that incorporate uncertainties.

References

[1] Arisumi, H., Yokoi, K., and Komoriya, K. Casting manipulationmidair control of a gripper by impulsive force. *Robotics, IEEE Transactions on 24*, 2 (2008), 402–415.

[2] Armstrong-Hlouvry, B., Dupont, P., and Wit, C. C. D. A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica 30*, 7 (1994), 1083 – 1138.

[3] Baker, B., Fortune, S., and Grosse, E. Stable prehension with a multi-fingered hand. *Proceedings. 1985 IEEE International Conference on Robotics and Automation 2* (1985), 570–575.

[4] Bicchi, A., and Sorrentino, R. Dexterous manipulation through rolling. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation* (May 1995), vol. 1, pp. 452–457 vol.1.

[5] Bowden, F., and Tabor, D. *Friction: An Introduction to Tribology.* Science study series. R.E. Krieger Publishing Company, 1973.

[6] Brock, D. Enhancing the dexterity of a robot hand using controlled slip. In *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on* (Apr 1988), pp. 249–251 vol.1.

[7] Chavan-Dafle, N., and Rodriguez, A. Prehensile pushing: In-hand manipulation with push-primitives. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on* (Sept 2015), pp. 6215–6222.

[8] Chavan-Dafle, N., and Rodriguez, A. Stable prehensile pushing: In-hand manipulation with alternating sticking contacts. In *ICRA* (2018).

[9] Chavan-Dafle, N., Rodriguez, A., Paolini, R., Tang, B., Srinivasa, S., Erdmann, M., Mason, M. T., Lundberg, I., Staab, H., and Fuhlbrigge, T. Extrinsic dexterity: In-hand manipulation with external forces. In *IEEE International Conference on Robotics and Automation (ICRA)* (May 2014).

[10] Chen, I.-M., and Burdick, J. A qualitative test for n-finger force-closure grasps on planar objects with applications to manipulation and finger gaits. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on* (May 1993), pp. 814–820 vol.2.

[11] Cherif, M., and Gupta, K. K. Global planning for dexterous reorientation of rigid objects: Finger tracking with rolling and sliding. *The International Journal of Robotics Research 20*, 1 (2001), 57–84.

[12] Cole, A., Hsu, P., and Sastry, S. Dynamic control of sliding by robot hands for regrasping. *Robotics and Automation, IEEE Transactions on 8*, 1 (Feb 1992), 42–52.

[13] COULOMB, C. A. Essai sur une application des regles de maximis et minimis a quelques problemes de statique relatifs a l'architecture. *Mem. Div. Sav. Acad.* (1773).

[14] Cutkosky, M. R., and Kao, I. Computing and controlling compliance of a robotic hand. *IEEE Transactions on Robotics and Automation 5*, 2 (Apr 1989), 151–165.

[15] Drake, S., and Laboratory, C. S. D. *Using Compliance in Lieu of Sensory Feedback for Automatic Assembly.* PhD Thesis, MIT, 1978.

[16] Fearing, R. Simplified grasping and manipulation with dextrous robot hands. *Robotics and Automation, IEEE Journal of 2*, 4 (December 1986), 188–195.

[17] Furukawa, N., Namiki, A., Taku, S., and Ishikawa, M. Dynamic regrasping using a high-speed multifingered hand and a high-speed vision system. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on* (May 2006), pp. 181–187.

[18] Goswami, A., and Peshkin, M. A. Mechanical computation for passive force control. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on* (May 1993), pp. 476–483 vol.1.

[19] Goyal, S. *Planar sliding of a rigid body with dry friction: limit surfaces and dynamics of motion.* PhD thesis, Cornell University, 1989.

[20] Goyal, S., Ruina, A., and Papadopoulos, J. Planar sliding with dry friction part 1. limit surface and moment function. *Wear 143*, 2 (1991), 307–330.

[21] Goyal, S., Ruina, A., and Papadopoulos, J. Planar sliding with dry friction part 2. dynamics of motion. *Wear 143*, 2 (1991), 331–352.

[22] Han, L., Guan, Y. S., Li, Z. X., Shi, Q., and Trinkle, J. C. Dextrous manipulation with rolling contacts. In *Proceedings of International Conference on Robotics and Automation* (Apr 1997), vol. 2, pp. 992–997 vol.2.

[23] Han, L., and Trinkle, J. Dextrous manipulation by rolling and finger gaiting. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on* (May 1998), vol. 1, pp. 730–735 vol.1.

[24] Hanafusa, H., and Asada, H. Stable prehension of objects by the robot hand with elastic fingers. *Transactions of the Society of Instrument and Control Engineers 13*, 4 (1977), 370–377.

[25] Hong, J., Lafferriere, G., Mishra, B., and Tan, X. Fine manipulation with multifinger hands. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on* (May 1990), pp. 1568–1573 vol.3.

[26] Hou, Y., Jia, Z., Johnson, A. M., and Mason, M. T. Robust planar dynamic pivoting by regulating inertial and grip forces. In *THE 12 TH INTERNATIONAL WORKSHOP ON THE ALGORITHMIC FOUNDATIONS OF ROBOTICS* (December 2016), Springer.

[27] Howard, S., Zefran, M., and Kumar, V. On the 6×6 cartesian stiffness matrix for three-dimensional motions. *Mechanism and Machine Theory 33*, 4 (1998), 389 – 408.

[28] Howard, W. S., and Kumar, V. On the stability of grasped objects. *IEEE Transactions on Robotics and Automation 12*, 6 (Dec 1996), 904–917.

[29] Howe, R. D., Kao, I., and Cutkosky, M. R. The sliding of robot fingers under combined torsion and shear loading. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation* (Apr 1988), pp. 103–105 vol.1.

[30] Huang, S., and Schimmels, J. M. Admittance selection for force-guided assembly of polygonal parts despite friction. *Robotics, IEEE Transactions on 20*, 5 (2004), 817–829.

[31] Ji, X., and Xiao, J. Planning motion compliant to complex contact states. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on* (2001), vol. 2, pp. 1512–1517 vol.2.

[32] Kao, I., and Cutkosky, M. R. Quasistatic manipulation with compliance and sliding. *The International Journal of Robotics Research 11*, 1 (1992), 20–40.

[33] Kenneth. Salisbury, J. Kinematic and force analysis of articulate hands. *PhD Thesis, Stanford University* (1982).

[34] Kumar, V., Todorov, E., and Levine, S. Optimal control with learned local models: Application to dexterous manipulation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* (May 2016), pp. 378–383.

[35] Li, M., Bekiroglu, Y., Kragic, D., and Billard, A. Learning of grasp adaptation through experience and tactile sensing. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on* (Sept 2014), pp. 3339–3346.

[36] Li, Z., and Canny, J. Motion of two rigid bodies with rolling constraint. *IEEE Transactions on Robotics and Automation 6*, 1 (Feb 1990), 62–72.

[37] Li, Z., Hsu, P., and Sastry, S. Grasping and coordinated manipulation by a multifingered robot hand. *The International Journal of Robotics Research 8*, 4 (1989), 33–50.

[38] Lynch, K. M. The Mechanics of Fine Manipulation by Pushing. *ICRA* (1992).

[39] Lynch, K. M., and Mason, M. T. Stable Pushing: Mechanics, Controllability, and Planning. *IJRR* (1996).

[40] Lynch, K. M., and Park, F. C. *Modern Robotics*. Cambridge University Press, 2017.

[41] Ma, R. R., and Dollar, A. M. An underactuated hand for efficient finger-gaiting-based dexterous manipulation. In *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)* (Dec 2014), pp. 2214–2219.

[42] Mason, M. T., and Salisbury, Jr., J. K. *Robot Hands and the Mechanics of Manipulation*. MIT Press, Cambridge, MA, USA, 1985.

[43] Meeussen, W., Schutter, J. D., Bruyninckx, H., Xiao, J., and Staffetti, E. Integration of planning and execution in force controlled compliant motion. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Aug 2005), pp. 1217–1222.

[44] Moseley, H. *Illustrations of Mechanics*. 2009.

[45] Odhner, L. U., and Dollar, A. M. Stable, open-loop precision manipulation with underactuated hands. *The International Journal of Robotics Research* (2015).

[46] Olsson, H., strm, K., de Wit, C. C., Gfvert, M., and Lischinsky, P. Friction models and friction compensation. *European Journal of Control 4*, 3 (1998), 176 – 195.

[47] Omata, T., and Nagata, K. Planning reorientation of an object with a multifingered hand. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation* (1994), pp. 3104–3110 vol.4.

[48] Park, H., Park, J., Lee, D. H., Park, J. H., Baeg, M. H., and Bae, J. H. Compliance-based robotic peg-in-hole assembly strategy without force feedback. *IEEE Transactions on Industrial Electronics 64*, 8 (Aug 2017), 6299–6309.

[49] Rojas, N., and Dollar, A. Characterization of the precision manipulation capabilities of robot hands via the continuous group of displacements. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on* (Sept 2014), pp. 1601–1608.

[50] Rus, D. In-hand dexterous manipulation of piecewise-smooth 3-d objects. *The International Journal of Robotics Research 18*, 4 (1999), 355–381.

[51] Salisbury, J. K., and Craig, J. J. Articulated hands: Force control and kinematic issues. *The International Journal of Robotics Research 1*, 1 (1982), 4–17.

[52] Salisbury, J. K., and Roth, B. Kinematic and force analysis of articulated mechanical hands. *Journal of Mechanisms, Transmissions, and Automation in Design 105* (03 1983), 35–41.

[53] Sarkar, N., Yun, X., and Kumar, V. Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robots. *The International Journal of Robotics Research 13*, 1 (1994), 55–69.

[54] Schimmels, J. M., and Peshkin, M. A. Admittance matrix design for force-guided assembly. *IEEE Transactions on Robotics and Automation 8*, 2 (Apr 1992), 213–227.

[55] Schimmels, J. M., and Peshkin, M. A. Force-assembly with friction. *IEEE Transactions on Robotics and Automation 10*, 4 (Aug 1994), 465–479.

[56] Schlegl, T., and Buss, M. Hybrid closed-loop control of robotic hand regrasping. In *ICRA'98* (1998), pp. 3026–3031.

[57] Shi, J., Woodruff, J. Z., and Lynch, K. M. Dynamic in-hand sliding manipulation. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on* (Sept 2015), pp. 870–877.

[58] Shi, J., Woodruff, J. Z., Umbanhowar, P. B., and Lynch, K. M. Dynamic in-hand sliding manipulation. *IEEE Transactions on Robotics 33*, 4 (Aug 2017), 778–795.

[59] Sintov, A., and Shapiro, A. Swing-up regrasping algorithm using energy control. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* (May 2016), pp. 4888–4893.

[60] Sundaralingam, B., and Hermans, T. Relaxed-rigidity constraints: In-grasp manipulation using purely kinematic trajectory optimization. In *Proceedings of Robotics: Science and Systems* (Cambridge, Massachusetts, July 2017).

[61] Trinkle, J., and Hunter, J. A framework for planning dexterous manipulation. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on* (Apr 1991), pp. 1245–1251 vol.2.

[62] Viña B., F. E., Karayiannidis, Y., Pauwels, K., Smith, C., and Kragic, D. In-hand manipulation using gravity and controlled slip. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on* (Sept 2015), pp. 5636–5641.

[63] Viña B., F. E., Karayiannidis, Y., Smith, C., and Kragic, D. Adaptive control for pivoting with visual and tactile feedback. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* (May 2016), pp. 399–406.

[64] Watson, P., and Laboratory, C. S. D. *A Multidimensional System Analysis of the Assembly Process as Performed by a Manipulator*. Society of Manufacturing Engineers, 1976.

[65] Whitney, D. E. Quasi-static assembly of compliantly supported rigid parts. *Journal of Dynamic Systems, Measurement, and Control 104*, 1 (1982), 65–77.

[66] Xydas, N., and Kao, I. Modeling of contact mechanics and friction limit surfaces for soft fingers in robotics, with experimental results. *The International Journal of Robotics Research 18*, 9 (1999), 941–950.

[67] Yashima, M., Shiina, Y., and Yamaguchi, H. Randomized manipulation planning for a multi-fingered hand by switching contact modes. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on* (Sept 2003), vol. 2, pp. 2689–2694 vol.2.

[68] Yashima, M., and Yamaguchi, H. Dynamic motion planning whole arm grasp systems based on switching contact modes. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on* (2002), vol. 3, pp. 2492–2499.

[69] Yoshikawa, T., and Nagai, K. Manipulating and grasping forces in manipulation by multifingered robot hands. *Robotics and Automation, IEEE Transactions on 7*, 1 (Feb 1991), 67–77.

[70] Zhang, H., Tanie, K., and Maekawa, H. Dextrous manipulation planning by grasp transformation. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on* (Apr 1996), vol. 4, pp. 3055–3060 vol.4.

[71] Zhou, J., Paolini, R., Bagnell, J. A., and Mason, M. T. A convex polynomial force-motion model for planar sliding: Identification and application. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* (May 2016), pp. 372–377.

APPENDIX A

Proof of Theorem 1

Proof. Without loss of generality, we prove the theorem for the case $d_{\text{goal}} > 0$, as indicated in Figure 4.3. With friction uncertainty included, the friction coefficient is $\mu \in [\mu_0(1-\epsilon),\ \mu_0(1+\epsilon)]$. The object acceleration during sliding therefore satisfies $a_{o,\text{actual}} \in [a_o - a_\epsilon,\ a_o + a_\epsilon]$, where $a_o = \mu_0 f_N$ and $a_\epsilon = \mu_0 f_N \epsilon$.

To understand these conditions, we first discuss sliding distance error caused by the friction uncertainty. For the case when the friction coefficient is larger than the estimate, the object will undershoot d_{goal}. The maximum sliding distance error caused by the underestimated friction coefficient is the area of the dark orange triangle d_2 in Figure 4.3, and

$$d_2 = \frac{(a + a_f)a_\epsilon d_{\text{goal}}}{(a_f - a_o)(a + a_o + a_\epsilon)}. \tag{A.1}$$

For the case where the actual friction coefficient is smaller than the estimate, the object will slide more than d_{goal}. The maximum sliding distance error is the area of the dark green triangle d_3 in Figure 4.3, which can be expressed as

$$d_3 = \frac{(a + a_f)a_\epsilon d_{\text{goal}}}{(a_f - a_o)(a + a_o - a_\epsilon)}. \tag{A.2}$$

From Equations (A.1) and (A.2), the maximum sliding distance errors satisfy $d_2 < d_3$ for any $a_f > a_o$ regardless of what a_f is chosen. Therefore we focus on the conditions on d_3 to ensure that the sliding distance errors decrease within the given rate.

From Equation (A.2), when $a_f > a_o$, and with given a_o and a_ϵ, as a_f increases to infinity d_3 converges to its minimum value of $\frac{a_\epsilon d_{\text{goal}}}{a+a_o-a_\epsilon}$. Although it initially seems counter-intuitive, increasing the finger acceleration a_f decreases the overshoot error because the phase durations become smaller (Equation (4.5)). To prevent the object from sliding too far, the condition $d_3 < d_{\text{goal}}$ should be ensured. Therefore we have a lower bound for a:

$$\frac{a_\epsilon}{a + a_o - a_\epsilon} < 1 \;\Rightarrow\; a > \mu_0 f_N(2\epsilon - 1).$$

An upper bound on a ensures that the object sticks in the beginning and end modes when the friction is overestimated, and is expressed by $a < \mu_0 f_N(1-\epsilon)$. To ensure a feasible a exists, the upper and lower bounds of a should satisfy

$$\mu_0 f_N(2\epsilon - 1) < \mu_0 f_N(1-\epsilon) \;\Rightarrow\; \epsilon < 2/3.$$

Note that the decrease of the net sliding error from one iteration to the next is never better than $\frac{a_\epsilon}{a+a_o-a_\epsilon} = \frac{\mu_0 f_N \epsilon}{a+\mu_0 f_N(1-\epsilon)}$. This gives the lower bound on the feasible convergence rate ρ.

The range of the actual sliding distance can be written as $d_{\text{actual}} \in [d_{\text{goal}} - d_2,\ d_{\text{goal}} + d_3]$, and the error in sliding distance as $e = d_{\text{goal}} - d_{\text{actual}} \in [d_2, -d_3]$.

At each iteration, the error e_k from the previous iteration becomes new $(d_{\text{goal}})_k$ and is used to replan a sliding motion:

$$(d_{\text{goal}})_{k+1} = (d_{\text{goal}})_k - (d_{\text{actual}})_k \in [(d_2)_k\ ,\ (-d_3)_k]. \tag{A.3}$$

To have the error in the net sliding distance converge to zero at least as fast as ρ^k, we need to choose a maximum finger acceleration a_f such that $e_k \leq \rho(d_{\text{goal}})_k$. For the case of an overestimated friction coefficient, Equations (4.3) and (A.1) show that when

$a_f \geq \frac{\mu_0 f_N[\mu_0 f_N(\epsilon-1)\rho - a(\epsilon+\rho)]}{\mu_0 f_N(\epsilon\rho-\rho+\epsilon)-a\rho}$ we have

$$\frac{|(d_2)_k|}{|(d_{\text{goal}})_k|} \leq \frac{a_o + a - a_\epsilon}{a_o + a + a_\epsilon}\rho < \rho. \tag{A.4}$$

Similarly for the case of an underestimated friction coefficient, Equations (4.3) and (A.2) show that when $a_f \geq \frac{\mu_0 f_N[\mu_0 f_N(\epsilon-1)\rho - a(\epsilon+\rho)]}{\mu_0 f_N(\epsilon\rho-\rho+\epsilon)-a\rho}$ we have

$$\frac{|(d_3)_k|}{|(d_{\text{goal}})_k|} \leq \rho. \tag{A.5}$$

From Equations (A.1) and (A.2) we have $d_2 < d_3$ for any $a_f > a_o$. Therefore we only need to satisfy the a_f constraint that leads to Equation (A.4), which yields $a_f \geq \frac{\mu_0 f_N[\mu_0 f_N(\epsilon-1)\rho - a(\epsilon+\rho)]}{\mu_0 f_N(\epsilon\rho-\rho+\epsilon)-a\rho}$.

Combining Equations (A.3)-(A.5) gives $\left|\frac{(d_{\text{goal}})_{k+1}}{(d_{\text{goal}})_k}\right| < \rho$, which demonstrates that $|(d_{\text{goal}})_k|$ converges exponentially to zero as the number of iterations k increases. □

APPENDIX B

Details of Limit Surface Modeling

We introduce a local finger frame $\mathcal{F}'_i$ at the center of the finger, and since the frames $\mathcal{F}'_1$ and $\mathcal{F}'_2$ are coincident in this system, we use $\mathcal{F}'$ to represent the frame as shown in Figures 4.9 and B.1. The y'-axis is along the direction from the center of the object to the center of the finger. The distance from the CM of the object to the center of the fingers is $\gamma = \left| \ [x_{rf}, y_{rf}]^\mathsf{T} \ \right|$.

The contact pressures p_1 and p_2 over the patches are a function of the position on the patch. We have constraints on the pressure distributions to keep the object planar when subject to the spring force, but the exact shape of the pressure distributions is unknown. We chose to approximate the pressure distributions as either constant or linearly varying distributions, as these are the lowest-order models that can satisfy the force-moment balance constraints that ensure the planar motion of the object. We assume that modeling errors leading to execution error can be accommodated by our iterative replanning approach.

In particular, the pressure distribution p_1 is assumed to be constant over the contact patch, while p_2 must vary over the contact patch to achieve balance of forces and moments that would otherwise move the object out of the plane. We assume p_2 varies linearly in the y' direction, as shown in Figures 4.9 and B.1. The shape of the contact patches are assumed constant and independent of the contact pressures.

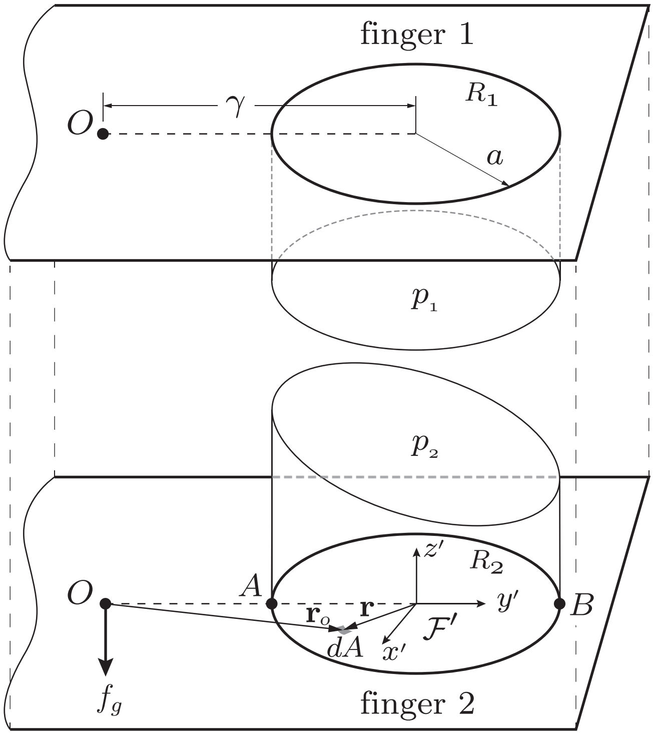


Figure B.1. Contact pressure distribution of the two-fingered pinch grasp.

B.1. Mechanics of the System

B.1.1. Finger Pressure Distributions

Since the contact pressure of finger 1 is evenly distributed, the total normal force of finger 1 f_{N1} is a function of the spring force S,

$$f_{N1} = \int_{R_1} p_1(\mathbf{r})\, dA = \frac{L_S}{L} S, \tag{B.1}$$

where R_1 is the circular contact region and dA is the infinitesimally small area located at $\mathbf{r} = [x_r, y_r]^\mathsf{T}$ with respect to the local frame $\mathcal{F}'$ as shown in Figure B.1. From Equation (B.1) we have

$$p_1(\mathbf{r}) = \begin{cases} \frac{L_S S}{\pi a^2 L} & \text{for } |\mathbf{r}| \leq a \\ 0 & \text{for } |\mathbf{r}| > a \end{cases}. \tag{B.2}$$

For finger 2 the contact pressure is assumed to be symmetrical about the y'-axis. Since we assume the shape of the pressure distribution function changes linearly in the y'-direction, as expressed by

$$p_2(\mathbf{r}) = \begin{cases} C_2 + ky_r & \text{for } |\mathbf{r}| \leq a \\ 0 & \text{for } |\mathbf{r}| > a \end{cases}, \tag{B.3}$$

where k is the change in pressure dp/dy_r, and C_2 is the constant term of p_2. The total normal force of finger 2 is

$$\begin{aligned} f_{N2} &= \int\limits_{R_2} p_2(\mathbf{r})\, dA = \int\limits_{-a}^{a} \int_{-\sqrt{a^2-y_r^2}}^{\sqrt{a^2-y_r^2}} (C_2 + ky_r) dx_r dy_r \\ &= \pi a^2 C_2. \end{aligned} \tag{B.4}$$

Since the motion of the system is in the xy plane of $\mathcal{W}$, the total force in the vertical direction and moments about the object CM must be balanced. We denote the total contact moments about the CM of the object as m_{ti}. So we have

$$m_{t1} = \int\limits_{R_1} [\mathbf{r}_o \times \hat{\mathbf{z}'}] p_1(\mathbf{r})\, dA = \frac{L_S S \gamma}{L}, \tag{B.5}$$

where $\mathbf{r}_o$ is the vector pointing from the object center O to the infinitesimally small area dA, $r_o = [x_r, y_r + \gamma]^\mathsf{T}$ as shown in Figure B.1, and $\hat{\mathbf{z}'}$ is the unit vector in $+z'$-direction.

For finger 2 we have

$$m_{t2} = \int_{R_2} [\mathbf{r}_o \times \hat{\mathbf{z}}'] p_2(\mathbf{r})\, dA = \frac{\pi a^2}{4}(a^2 k + 4C_2\gamma). \tag{B.6}$$

The force and moment balance equations of the object are:

$$f_{N1} + f_{N2} + f_g = 0, \tag{B.7}$$

$$m_{t1} + m_{t2} = 0. \tag{B.8}$$

Substituting Equations (B.1), (B.4), (B.5) and (B.6) into Equations (B.7) and (B.8) gives

$$C_2 = -\frac{1}{\pi a^2}\left(f_g + \frac{L_S S}{L}\right), \tag{B.9}$$

$$k = \frac{4 f_g \gamma}{\pi a^4}. \tag{B.10}$$

Substituting Equation (B.9) into (B.4) gives

$$f_{N2} = -\left(f_g + \frac{L_S S}{L}\right). \tag{B.11}$$

The finger contacts can only apply forces into the object, which means that for finger 2 the contact pressure at any contact point must be nonnegative, i.e., $\forall \mathbf{r} \in R_2 : p_2(\mathbf{r}) \geq 0$. From Equation (B.10), we have $k \leq 0$. And since $\gamma \geq 0$ and $f_g \leq 0$, the minimum contact pressure of finger 2 is the pressure at point B. To ensure feasible contact pressures, $p_2(\mathbf{r}_B) \geq 0$ should be satisfied, where $\mathbf{r}_B = [0, a]^\intercal$. By substituting Equations (B.3), (B.9) and (B.10) into the inequality $p_2(\mathbf{r}_B) \geq 0$, we can solve the maximum distance between the finger center and the object center as

$$\gamma \leq \frac{a}{4}\left(1 + \frac{L_S S}{f_g L}\right). \tag{B.12}$$

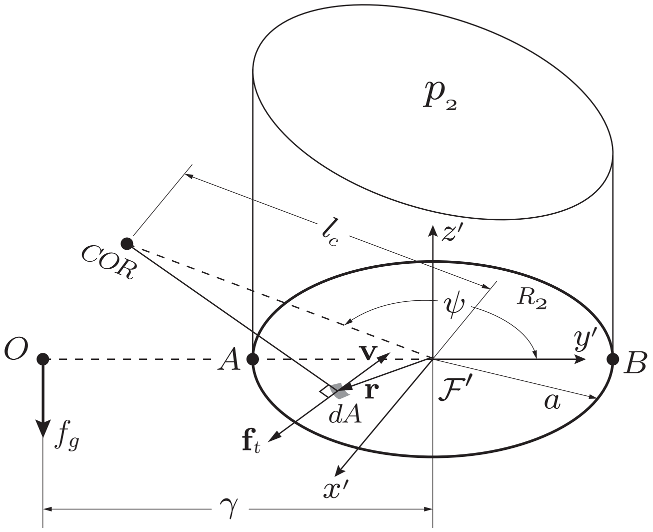


Figure B.2. Close up view of finger 2 pressure distribution.

When γ exceeds the limit, the contacts break and the object falls out of the grasp since the contacts cannot supply pulling forces. To maintain contacts in this situation, the spring force S should be increased.

B.2. Modeling of Limit Surfaces

B.2.1. Constructing Limit Surface Numerically

With the expressions for the finger contact pressure distributions given above, we calculate the frictional forces and moments caused by the contacts for different sliding directions. The set of possible relative velocities at a contact can be parameterized using the center of rotation (COR) formulation as in [**66**]. The finger's instantaneous center of rotation is defined by two variables: the distance to the center of the finger l_c and the angle from the y'-axis ψ, as shown in Figure B.2. The relative sliding velocity vector at the infinitesimal area dA is $\mathbf{v}$. The tangential frictional force $\mathbf{f}_t$ acts in the opposite direction of $\mathbf{v}$.The total tangential frictional force of each contact is calculated by integrating shear forces over the entire contact patch R_i as

$$\mathbf{f}_{t,i} = \begin{bmatrix} f_{x,i} \\ f_{y,i} \end{bmatrix} = -\int\limits_{R_i} \mu \hat{\mathbf{v}} p_i(\mathbf{r}) dA, \tag{B.13}$$

and the total frictional moment about the z'-axis is

$$m_{z,i} = -\int\limits_{R_i} \mu [\mathbf{r} \times \hat{\mathbf{v}}] p_i(\mathbf{r}) dA, \tag{B.14}$$

where $\hat{\mathbf{v}}$ is the unit vector in the direction of $\mathbf{v}$:

$$\hat{\mathbf{v}} = \frac{1}{\Lambda} \begin{bmatrix} l_c \cos\psi - y_r \\ l_c \sin\psi + x_r \end{bmatrix}. \tag{B.15}$$

where $\Lambda = \sqrt{l_c^2 - 2y_r l_c \cos\psi + 2x l_c \sin\psi + x_r^2 + y_r^2}$. Substituting Equation (B.15) into Equations (B.13) and (B.14), we have

$$\mathbf{f}_{t,i} = \int\limits_{-a}^{a} \int_{-\sqrt{a^2-y_r^2}}^{\sqrt{a^2-y_r^2}} \frac{-\mu p_i(\mathbf{r})}{\Lambda} \begin{bmatrix} l_c \cos\psi - y_r \\ l_c \sin\psi + x_r \end{bmatrix} dx_r dy_r, \tag{B.16}$$

$$m_{z,i} = \int\limits_{-a}^{a} \int_{-\sqrt{a^2-y_r^2}}^{\sqrt{a^2-y_r^2}} \frac{-\mu \Phi p_i(\mathbf{r})}{\Lambda} dx_r dy_r\,, \tag{B.17}$$

where $\Phi = l_c x_r \sin\psi - l_c y_r \cos\psi + x_r^2 + y_r^2$.

The frictional forces and moments of finger 1 can be considered as a special case of finger 2, where $\gamma = 0$ and both p_1 and p_2 are constant. Therefore, we only analyze the limit surface for finger 2.

Equations (B.16) and (B.17) do not have closed-form solutions since they are elliptic integrals. Equations (B.9) and (B.10) can be numerically integrated to construct the limit surfaces. Figures B.3 and B.4 show the results of numerical integration of limit surfaces

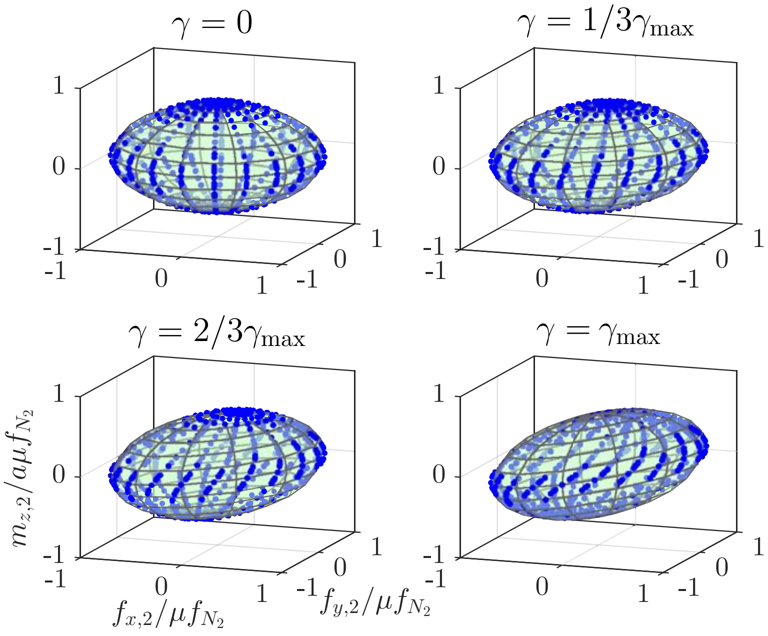


Figure B.3. Numerically integrated and approximated limit surfaces of finger 2. The axes of the friction force space are aligned with the local frame $\mathcal{F}'$ and normalized. Blue dots are the numerical integration results, green ellipsoids show the approximated limit surfaces.

of finger 2 with four different values of γ. Each blue dot represents an integration result of a COR position on the $x'y'$-plane (a pair value of l_c and ψ). Substituting $L_S = 0.05\,\text{m}$, $L = 0.17\,\text{m}$, $f_g = -0.5\,\text{N}$, $S = -7\,\text{N}$, $a = 0.0254\,\text{m}$ into Equation (B.12) we find the maximum γ is $\gamma_{\max} \approx 1.279\,a$.

B.2.2. Approximation of the Limit Surfaces

Since the shape of LS_1 is a special case of LS_2 when $\gamma = 0$, in this subsection we focus on the derivation of the expression for the approximated LS_2. The idea is to fit an ellipsoid to the numerical integrals in the local frame $\mathcal{F}'$ that deforms as γ increases. This LS approximation is then expressed in the local frame $\mathcal{F}^+$ which is used in the dynamics derived in Section 4.3.

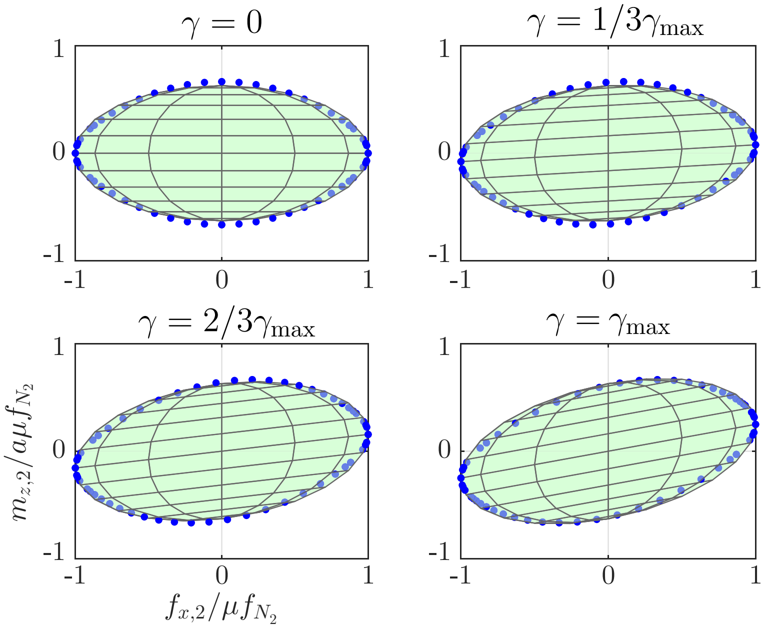


Figure B.4. Numerically integrated and approximated limit surfaces when the COR is moving along the y'-axis, shown in the $f_x m_z$-plane. Blue dots are numerical integration results, green ellipsoids show the approximated limit surfaces.

Observation 1. *From Figures B.3 and B.4 we observe that as the distance γ from the object center to the finger center increases, the m_z components of points on the limit surface increase or decrease by a factor linear in both γ and f_x.*

From Equation (B.11) the total normal force is not affected by γ, which means that the maximum linear frictional force μf_{N_2} will be the same as long as $\gamma \leq \gamma_{\max}$, and the projection of the limit surfaces to the $f_x f_y$-plane will be the same circle centered at the origin with radius μf_{N_2}. Equation (B.10) shows that the contact pressure distribution is determined by γ, which also affects the maximum frictional moment $m_{z,2}$ at each COR. Therefore as γ changes, the shape of the limit surfaces changes in the m_z-direction.

Based on observation 1, we use deformed ellipsoids to approximate the limit surfaces. We denote $\mathbf{f}_e^{\mathcal{F}'} = [f_{x,e}, f_{y,e}, m_{z,e}]^\intercal$ as an arbitrary vector on an ellipsoid centered at the origin of the local finger frame $\mathcal{F}'$, and $\mathbf{f}^{\mathcal{F}'} = [f_x, f_y, m_z]^\mathsf{T}$ as an arbitrary vector on the corresponding approximated limit surface. The ellipsoid is represented by

$$(\mathbf{f}_e^{\mathcal{F}'})^\mathsf{T}\mathbf{A}_e\mathbf{f}_e^{\mathcal{F}'} = 1, \tag{B.18}$$

where the matrix $\mathbf{A}_e \in \mathbb{R}^{3\times 3}$ is a symmetric positive-definite matrix that determines the shape of the ellipsoid. In the general ellipsoid definition, $\mathbf{A}_e = \mathrm{diag}(s_1^{-2}, s_2^{-2}, s_3^{-2})$ where s_1, s_2 and s_3 represent the lengths of the semi-principal axes. We again assume isotropic dry friction so the maximum tangential force the contact can resist is $s_1 = s_2 = \mu|f_{N_i}|$. The maximum moment along the normal direction is $s_3 = ca\mu|f_{N_i}|$, where a is the radius of the contact patch and c is a constant from numerical integration. Here we take $c = 0.63$ based on the findings in [**66**].

Since the limit surface is approximated by the ellipsoid deformed linearly in the m_z-direction proportional to f_x, we have

$$m_z = m_{z,e} + \kappa(\gamma)f_{x,e}, \tag{B.19}$$

where $\kappa(\gamma)$ is a variable which determines the linear mapping.

To derive $\kappa(\gamma)$, we choose a critical point $\mathbf{f}_*^{\mathcal{F}'} = [f_x^*, f_y^*, m_z^*]^\mathsf{T}$ in the frame $\mathcal{F}'$. Let $f_x^* = \mu f_{N_2}, f_y^* = 0$ so the projection of $\mathbf{f}_*^{\mathcal{F}'}$ in the f_xf_y-plane is at the edge of the limit circle. At this point, from the ellipsoid definition, we have $m_{z,e}^* = 0$. From Equation (B.19) we find

$$\kappa(\gamma) = m_z^*/f_{x,e}^*. \tag{B.20}$$

We calculate m_z^* from Equations (B.10) and (B.17) by substituting $\psi = 0$ and $l_c = -\infty$:

$$\begin{aligned} m_z^* &= -\mu \int_{-a}^{a} \int_{-\sqrt{a^2-y_r^2}}^{\sqrt{a^2-y_r^2}} y_r(C_2 + ky_r)dx_r dy_r \\ &= -\frac{\mu\pi a^4 k}{4} = -\mu f_g \gamma. \end{aligned} \tag{B.21}$$

Since the ellipsoid is only deformed in the m_z-direction, we have $f_{x,e}^* = f_x^* = \mu f_{N_2}$. Substituting this expression and Equation (B.21) into Equation (B.20) we have

$$\kappa(\gamma) = \frac{m_z^*}{f_x^*} = -\frac{f_g \gamma}{f_{N_2}}. \tag{B.22}$$

The transformation from the ellipsoid to the limit surface is given by

$$(\mathbf{f}^{\mathcal{F}'})^{\mathsf{T}} = (\mathbf{f}_e^{\mathcal{F}'})^{\mathsf{T}} \mathbf{D}, \tag{B.23}$$

where $\mathbf{D}$ is an affine transformation matrix which deforms the ellipsoid as

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & \kappa \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

From Equations (B.18) and (B.23) we have $(\mathbf{f}^{\mathcal{F}'})^{\mathsf{T}} \mathbf{D}^{-1} \mathbf{A}_e \mathbf{D}^{-\mathsf{T}} \mathbf{f}^{\mathcal{F}'} = 1$. The expression for all the points on the limit surface can be written as

$$(\mathbf{f}^{\mathcal{F}'})^{\mathsf{T}} \mathbf{A}_m \mathbf{f}^{\mathcal{F}'} = 1, \tag{B.24}$$

where

$$\mathbf{A}_m = \mathbf{D}^{-1} \mathbf{A}_e \mathbf{D}^{-\mathsf{T}} = \begin{bmatrix} (\kappa^2/s_3^2 + s_1^{-2}) & 0 & -\kappa/s_3^2 \\ 0 & s_2^{-2} & 0 \\ -\kappa/s_3^2 & 0 & s_3^{-2} \end{bmatrix}.$$

To describe the frictional limit surface in the frame $\mathcal{F}^+$, we have

$$\mathbf{f}^{\mathsf{T}} = (\mathbf{f}^{\mathcal{F}'})^{\mathsf{T}}\mathbf{R}^{\mathcal{F}'\mathcal{F}^+}, \tag{B.25}$$

where $\mathbf{R}^{\mathcal{F}'\mathcal{F}^+}$ is a transformation matrix that transfers the reference frame of linear force vectors from $\mathcal{F}'$ to $\mathcal{F}^+$, and

$$\mathbf{R}^{\mathcal{F}'\mathcal{F}^+} = \begin{bmatrix} \sin\phi & -\cos\phi & 0 \\ \cos\phi & \sin\phi & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where $\phi = \tan^{-1}(\frac{y_{rf2}}{x_{rf2}})$ as shown in Figure 4.5. From Equations (B.24) and (B.25), we write the equation of the limit surface in frame $\mathcal{F}^+$ as

$$\mathbf{f}^{\mathsf{T}}\mathbf{A}\mathbf{f} = 1, \tag{B.26}$$

where $\mathbf{A} = (\mathbf{R}^{\mathcal{F}'\mathcal{F}^+})^{-1}\mathbf{A}_m(\mathbf{R}^{\mathcal{F}'\mathcal{F}^+})^{-\mathsf{T}}$.

Comparisons showing the good agreement between the approximated frictional limit surface and the numerically integrated points are shown in Figures B.3 and B.4.