

Manipulation with Shared Grasping

Yifan Hou
Carnegie Mellon University
yifanh@cmu.edu

Zhenzhong Jia
Southern University of Science and Technology
jjazz@sustech.edu.cn

Matthew T. Mason
Carnegie Mellon University
mattmason@cmu.edu

Abstract—A *shared grasp* is a grasp formed by contacts between the manipulated object and both the robot hand and the environment. By trading off hand contacts for environmental contacts, a shared grasp requires fewer contacts with the hand, and enables manipulation even when a full grasp is not possible. Previous research has used shared grasps for non-prehensile manipulation such as pivoting and tumbling. This paper treats the problem more generally, with methods to select the best shared grasp and robot actions for a desired object motion. The central issue is to evaluate the feasible contact modes: for each contact, whether that contact will remain active, and whether slip will occur. Robustness is important. When a contact mode fails, e.g., when a contact is lost, or when unintentional slip occurs, the operation will fail, and in some cases damage may occur. In this work, we enumerate all feasible contact modes, calculate corresponding controls, and select the most robust candidate. We can also optimize the contact geometry for robustness. This paper employs quasi-static analysis of planar rigid bodies with Coulomb friction to derive the algorithms and controls. Finally, we demonstrate the robustness of shared grasping and the use of our methods in representative experiments and examples. The video can be found at <https://youtu.be/tyNhJvRYZNk>

I. INTRODUCTION

Grasping is widely adopted in automation industry because of its robustness, especially when compared with non-prehensile manipulative actions. A force-closure grasp can neutralize disturbance forces in any direction up to a magnitude determined by the gripping force. When grasping locations are not available on the object, the robot can only move the object with the help of external force resources [7], e.g. pivot or tumble an object on a table [1, 38, 30]. Unlike grasping, these are usually non-prehensile manipulations which are less robust against disturbance forces.

Salisbury [29] introduced the idea of “whole arm manipulation”, suggesting using all available surfaces on the robot for more ways to manipulate an object. Following this thought, if we treat the environment as another finger of the robot, manipulation under external contacts is like grasping from the object’s point of view. We call it *shared grasping*. With shared grasping, the object can stay in a force closure even when no force closure grasps are available. Under suitable contact

modes, the hand can move the object while maintaining the force closure, which opens possibilities for robust manipulation beyond grasping alone. This view motivates us to design a more general framework for manipulation under external contacts that can:

- Analyze the feasibility and robustness of a shared grasping system.
- Find robust robot actions to achieve a desired object motion and/or contact mode.

Although grasping and shared grasping are conceptually similar, traditional analyses for grasping stability do not work for shared grasping. In grasping, all contact points are assumed to be sticking [26], while in shared grasping all contact modes are considered to allow for more motion possibilities. With multiple possible contact modes, traditional force-based grasping stability analysis [32] cannot distinguish some contact modes from each other.

To overcome the mode ambiguity, and also to control the exact motion of the shared-grasped object, our approach models the robot action by *Hybrid Force-Velocity Control (HFVC)*. Force control and velocity control are special cases of HFVC. We use the velocity portion of HFVC to filter out undesired contact modes and maintain the desired object velocity. Note that the traditional force-based grasping analysis cannot model HFVC properly, because the forces in the velocity-controlled directions are not controllable.

In this work, we provide a method to analyze the quasi-static stability of planar shared grasping systems in any contact mode under HFVC. The foundation of our stability analysis is a distinguishable cone representation for the object wrenches from different contact modes. In the object wrench space, we illustrate how the change of robot action and model parameters can cause mode transitions, so as to quantify the *stability margin*, i.e. how much disturbance a contact mode can handle. To get around the force indeterminacy in velocity-controlled directions, we project the wrench space onto the force-controlled subspace and pick a force control there. This is why we call our approach *wrench stamping*.

During the computation of stability margin, wrench stamping also computes a robust HFVC action to achieve the desired object motion.

We demonstrate how to use wrench stamping in two problems: 1) How to select a contact mode for a multi-contact manipulation task; 2) Given a contact mode, how to choose finger placement locations on the object. In the first problem, we pick the contact mode with highest stability margin. In the second problem, we compute finger locations by maximizing the stability margin for the given mode. This is possible because we can compute the analytical gradient of the stability margin about the contact geometry parameters. Note that we can also optimize over the environmental contact locations and friction coefficients, which could be useful for system design in industrial applications.

The paper is organized as follows. After introducing related work in Section II, we first analyze the feasibility of contact modes in a shared grasping system in Section III. Based on the analysis, we derive the stability margin for any contact mode in Section IV. Section V uses our wrench stamping to solve mode selection and contact point optimization. We provide experimental results in several representative shared grasping problems in Section VI, and conclude with a discussion on limitations and future work in Section VII.

II. RELATED WORK

A. Manipulation Modeling with External Contacts

There is lot of prior work on manipulation with external contacts. By pressing a grasped object against a support, the robot can do in-hand manipulation [33, 12, 15, 4] or transport the object without bearing its full weight [1, 38]. Without a grasp, a finger can still move an object by planar pushing [19, 41, 11], pivoting or tumbling [30, 14]. Daffle et al. [7] demonstrated that a simple robot hand can do many tasks by sequencing multiple open loop actions with external contacts.

Several of these examples were prehensile, such as moving a block on a surface by rotating the block about one of its corners [1, 38] or edges on the surface [30]. The object is in a force closure formed by two point-finger contacts and a point or edge table contact. These works computed the conditions of maintaining static balance throughout the motion, including minimal friction coefficient (all contacts were sticking) and range of finger locations. Planar pushing can also be prehensile, examples are rotating a paper card on a table by pressing on it with two fingers [17], and in-hand manipulation with a parallel gripper by external contacts [4, 6].

B. Robust Control of Contact-Rich Manipulation

Manipulation control under external contacts is difficult because a control designed for one contact mode can result in another mode. Previous work on manipulation control were mainly concerned with switching to the right controller for the right mode. It is popular to do mode scheduling (either planned or hand-coded) before computing a feedback controller [11, 10].

Using hybrid force-velocity control (HFVC, also called hybrid 'force-position' or 'position-force' control) brings robustness to manipulation, because the force control and velocity control can handle model uncertainties and force disturbances, respectively. There is a long thread of work on HFVC for manipulation in a rigid environment [20, 39, 28, 37], and stability analysis for the whole robot-contact system under HFVC [22]. Uchiyama and Dauchez [35] computes HFVC for manipulating an object with two fingertips.

Hou and Mason [14] considered the case when the object may have multiple contacts with both the robot hand and the environment. They introduced the hybrid servoing algorithm to compute a HFVC action, and provided criteria to evaluate the robustness of a multi-contact system against several kinds of failures [13]. This work extends previous work by considering all possible contact modes, as well as proposing a simpler robustness score with clear physical meanings.

C. Motion Planning with Contacts

The model of rigid body contacts is discontinuous. Gradients do not exist at the making or breaking of contacts, thereby making powerful continuous solvers unusable. A popular way around is to model the contacts by complementary constraints on continuous variables [27]. However, these constraints make the optimization problem close to ill-condition and sensitive to the choice of initial trajectory. Another direction is to approximate the contact dynamics with continuous models [23, 24, 34]. So far there have been limited success [25] in transferring motions planned with simplified dynamics into experiments. Sampling based planning methods handle discrete states naturally. They find solutions by solving steering problems under contact constraints [36, 5], or by projecting the sampled motion into the constrained manifold [2].

A key to successful experiments is the ability to handle uncertainties. Object pose uncertainties can be reduced by contacts [18, 42, 31]. However, there is little work on making the contact modes robust against modeling uncertainties and force disturbances. This paper lays the foundation for such planning by providing fast solutions for robustness-based mode selection and contact point selection.

In recent years, learning based methods have also enabled robots to do manipulation tasks with external contacts, such as opening doors [9] and pushing in clutter for grasping [8, 40]. However, these learned tasks are intrinsically stable: temporarily falling into a bad contact mode will not fail the task. We have not seen work on learning shared grasping tasks such as pivoting or tumbling. The reason could be the difficulty in simulating contacts and the inefficiency/danger of action exploration under holonomic constraints.

III. FEASIBILITY OF A CONTACT MODE

To begin with, in this section we derive the mechanical condition under which a contact mode is feasible and unique. This section lies the base for our robustness analysis in the next section. First, we introduce the symbols.

Consider a robot hand, a rigid object and a rigid environment. Denote H, O, W as the hand frame, object frame and world inertia frame, respectively. We use the symbol ${}^A c_B$ to describe the quantity c of frame B measured in frame A , where c could be position p , normal n , force f , etc. Denote N_D as the degree-of-freedom of a rigid body, N_d as the dimension of space ($N_d = 2$ for planar systems). We make the following assumptions:

- The system is quasi-static: inertia forces are ignored.
- Friction obeys Coulomb's Law; the maximum stiction equals the sliding friction.
- The hand joints are locked.
- Gravity is not significant comparing to contact forces.

A description of a shared grasping system includes:

- Robot hand pose and object pose.
- A list of contact points and normals on the object.
- Coulomb friction coefficient μ for each contact.

A. Newton's Law for Wrench Space Analysis

Define the generalized velocity vector V and force vector F of the system in the robot hand frame H :

$$V = \begin{bmatrix} {}^H v_O \\ {}^H v_H \end{bmatrix}, F = \begin{bmatrix} {}^H f_O \\ {}^H f_H \end{bmatrix}. \quad (1)$$

${}^H f_H$ is the robot control force, ${}^H f_O$ is always zero since the object is free. We choose the hand frame because it is easier to describe HFVC later. The Newton's Second Law for the hand-object system takes the form:

$$\begin{bmatrix} \mathbf{J}_e^T & -\mathbf{J}_h^T \\ 0 & \mathbf{J}_h^T \end{bmatrix} \begin{bmatrix} \tau_e \\ \tau_h \end{bmatrix} + \begin{bmatrix} {}^H f_O \\ {}^H f_H \end{bmatrix} = 0. \quad (2)$$

where $\tau_e, \tau_h \geq 0$ are vectors of contact wrench magnitudes; each row of the contact Jacobian $\mathbf{J}_e$ and $\mathbf{J}_h$ represents the wrench of a friction cone edge of an environment contact and hand contact, respectively. The derivation of the contact Jacobians can be found in [26]. Each contact point contributes up to two rows to the contact Jacobian, depending on its mode:

- (s) Separation: no wrench from the contact;
- (f) Fixed: wrenches for both the left and right edges;
- (l) Left sliding: only the wrench for the right edge;
- (r) Right sliding: only the wrench for the left edge.

Following the convention in [21], we use a string of the abbreviation letters to describe a contact mode, e.g. 'ffsl'.

B. A Distinguishable Cone Representation for Modes

There are many ways to check the feasibility of (2), i.e. whether a set of feasible forces exist. It is well-known that all the contact wrenches on the object forms a polyhedral convex cone (PCC) in the object wrench space; if the cone spans the whole space, the contacts on the object are in static balance and the object is in force closure [26]. This way of using PCC is not helpful for shared grasping, because it cannot distinguish different contact modes and every force closure contact mode looks the same (span the whole wrench space). We must modify the representation.

Rewrite equation (2) using ${}^H f_O = 0$:

$$\mathbf{J}_e^T \tau_e = \mathbf{J}_h^T \tau_h = -{}^H f_H. \quad (3)$$

This equation maps the environmental contact force τ_e and the hand contact force τ_h into the generalized force space of the hand, which is the space where the robot action ${}^H f_H$ lives in. Since $\tau_e \geq 0, \tau_h \geq 0$, $\mathbf{J}_e^T \tau_e$ and $\mathbf{J}_h^T \tau_h$ each again forms a polyhedral convex cone in this wrench space, call them the *environment cone* $\mathbb{C}_e$ and the *hand cone* $\mathbb{C}_h$:

$$\begin{aligned} \mathbb{C}_e &= \{\mathbf{J}_e^T \tau_e \mid \tau_e \geq 0\} \\ \mathbb{C}_h &= \{\mathbf{J}_h^T \tau_h \mid \tau_h \geq 0\} \end{aligned} \quad (4)$$

An example is shown in Figure 1. Rows of $\mathbf{J}_e$ and $\mathbf{J}_h$ are

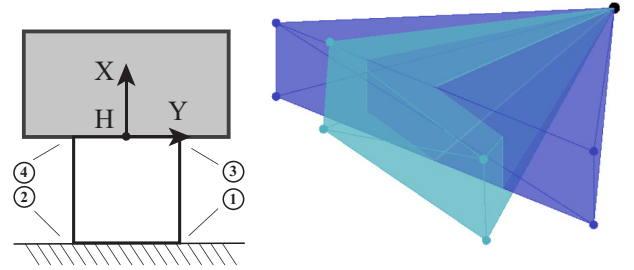


Fig. 1: Left: a shared grasping system with a cube object (bottom) and a robot palm (top). The circled numbers show the ordering of contact points. Right: the hand cone (purple) and environment cone (blue) for the "ffff" mode.

generators of the cones. Our condition of force balance is:

The existence of a nonzero intersection between $\mathbb{C}_e$ and $\mathbb{C}_h$ is equivalent to the existence of a non-trivial (all zero) solution $(\tau_e, \tau_h, {}^H f_H)$ to the static balance equation (3).

Now we explain how this new intersection-based condition makes a difference in distinguishing contact modes. We denote this intersection as $\mathbb{C}_m$ and call it the *cone of the mode m*. We can draw $\mathbb{C}_m$ for all the modes in the same space, as in Figure 2. We use the contact mode enumeration algorithm by Huang et al. [16] to get a list of modes. Note that these cones

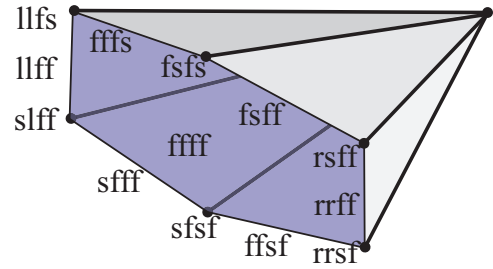


Fig. 2: A closer look at the cones of the example in Figure 1 with annotation. In this example, each non-zero face of the 3D cone is the cone of a contact mode.

are distinguishable, although most contact modes have force closure. This is because different contact modes always have different contact wrenches.

From (3) we know the robot can directly apply a wrench $w = -^H f_H$ in this wrench space, then as long as

$$w \in \mathbb{C}_m, \quad (5)$$

mode m is feasible under robot action w . We can make the claim stronger:

If the robot action w is contained only in $\mathbb{C}_m$, the system must be in mode m .

Now it seems straightforward to enforce a contact mode m : pick a robot wrench w in $\mathbb{C}_m$ while staying away from the cones of other modes. The system will be in mode m if such w exists. However, this approach lacks robustness because the cones of different modes are not well separated from each other. For example, in Figure 2 every facet of the “ffff” mode denotes another mode. If we apply w in one of those modes, a tiny disturbance force could push the actual applied wrench into the “ffff” mode. This shows a limitation of purely force-based method in multi-contact manipulation. We show how to overcome it in section III-C shortly.

Before we move on, another thing we can learn is the range of wrenches the robot may apply. Denote $\mathbb{U}_{AF}$ as the union of the hand cone and environment cone of the all-fixed mode. Denote $\mathbb{W}$ as the whole N_D dimensional wrench space. We can tell the result of applying a wrench w as follows.

- $w \in \mathbb{C}_{AF}$: the object can have static force balance;
- $w \in \mathbb{U}_{AF} - \mathbb{C}_{AF}$: the contact forces cannot balance the wrench, the object is accelerating;
- $w \in \mathbb{W} - \mathbb{U}_{AF}$: (impossible) some contact forces are outside of their friction cones.

Depending on the existence of $\mathbb{C}_m$, we call mode m *F-feasible* or *F-infeasible*.

C. Feasibility under Hybrid Force-Velocity Control(HFVC)

In this subsection we explain how to uniquely select one mode by replacing pure force control w with HFVC. HFVC decompose the robot action space into a velocity control subspace and force control subspace. Denote n_{af} and n_{av} as the dimension of force and velocity control subspace in HFVC, $n_{af} + n_{av} = N_d$. Define transformation $\mathbf{R}_a \in \mathbb{R}^{N_d \times N_d}$, we can describe a HFVC by:

$$\begin{bmatrix} \omega_{af} \\ \omega_{av} \end{bmatrix} = \mathbf{R}_a^H v_H, \quad \begin{bmatrix} \eta_{af} \\ \eta_{av} \end{bmatrix} = \mathbf{R}_a^H f_H, \quad (6)$$

where $\eta_{af} \in \mathbb{R}^{n_{af}}$ and $\omega_{av} \in \mathbb{R}^{n_{av}}$ are the force and velocity control, respectively. η_{av} , ω_{af} denote the uncontrollable variables: force/velocity in the velocity/force-controlled direction. The HFVC imposes a velocity constraint:

$$\begin{bmatrix} 0 & \mathbf{R}_a^{(n_{av})} \end{bmatrix} V = \omega_{av}. \quad (7)$$

$\mathbf{R}_a^{(n_{av})}$ is the last n_{av} rows of $\mathbf{R}_a$. Denote $\mathbf{C}_v = [0 \ \mathbf{R}_a^{(n_{av})}]$ and write the velocity constraint as $\mathbf{C}_v V = \omega_{av}$.

Given a desired contact mode, we use hybrid servoing [14] to compute the force-velocity decomposition $\mathbf{R}_a$ and velocity control ω_{av} . Hybrid servoing quickly finds a velocity control that is compatible with the contact constraints. We don't use

the second half of hybrid servoing which computes the force control η_{af} , because the original algorithm was not designed for handling sliding contacts. The inputs to hybrid servoing include a description of the goal velocity:

$$\mathbf{G}V = b_G, \quad (8)$$

and contact constraints on velocity $\mathbf{N}_m V = 0$, as explained below. In the following we analyze the system under a HFVC and use a new approach to compute the force control η_{af} .

1) *Model filtering by velocity control*: Denote the velocity constraint imposed by the contacts in contact mode m by:

$$\begin{aligned} \mathbf{N}_m V &= 0 \\ \mathbf{M}_m V &\leq 0, \end{aligned} \quad (9)$$

including non-penetration constraint, sticking constraint and sliding direction constraint. We call mode m **V-feasible** if the velocity constraints (7) and (9) have a solution, **V-infeasible** otherwise.

Before we can safely ignore all the V-infeasible modes, we need to consider the possibility of *crashing* [13], i.e. the situation where a small robot motion causes huge internal force. Crashing can be dangerous and must be avoided. In crashing, the system can remain in static balance when the force in the velocity-controlled directions becomes arbitrarily large. In other words, $\mathbb{C}_m$ must intersect the set of positive forces in the velocity-controlled directions, denoted as $\mathbb{C}_v$:

$$\mathbb{C}_m \cap \mathbb{C}_v \neq \{0\}. \quad (10)$$

This is a necessary condition. $\mathbb{C}_v$ here can be derived from (4) and (6):

$$\mathbb{C}_v = \{-\mathbf{R}_{av}^{-1} \tau_v | \tau_v \geq 0\}, \quad (11)$$

where $\mathbf{R}_{av}^{-1}$ represents the last n_{av} columns of $\mathbf{R}_a^{-1}$. $\mathbb{C}_v$ is again a PCC, we call it the *velocity cone*.

Remember $\mathbb{C}_{AF}$ contains all $\mathbb{C}_m$. Using (10) we can derive the sufficient condition for crashing-avoidance:

$$\mathbb{C}_{AF} \cap \mathbb{C}_v = \{0\}, \quad (12)$$

This condition ensures no crashing in all possible modes. If it is satisfied, we can safely ignore all the V-infeasible modes, which makes it easier to separate the desired mode from the rest. This is illustrated in the middle figure of Figure 3.

D. Mode Selection by Force Control

There is usually more than one V-feasible mode. To secure the desired mode, the robot still needs to apply suitable force. Equation (6) and (3) gives:

$$\mathbf{J}_e^T \tau_e = \mathbf{J}_h^T \tau_h = w = -\mathbf{R}_a^{-1} \begin{bmatrix} \eta_{af} \\ \eta_{av} \end{bmatrix}, \quad (13)$$

so the force-velocity decomposition happens directly in the wrench space where the hand and environmental cones live.

Note again that we can *not* control η_{av} , so we cannot directly place the wrench vector w in the cone of the desired mode. To resolve the difficulty, we project all V-feasible cones $\mathbb{C}_m$ into the force-controlled subspace. Those projections $\bar{\mathbb{C}}_m$

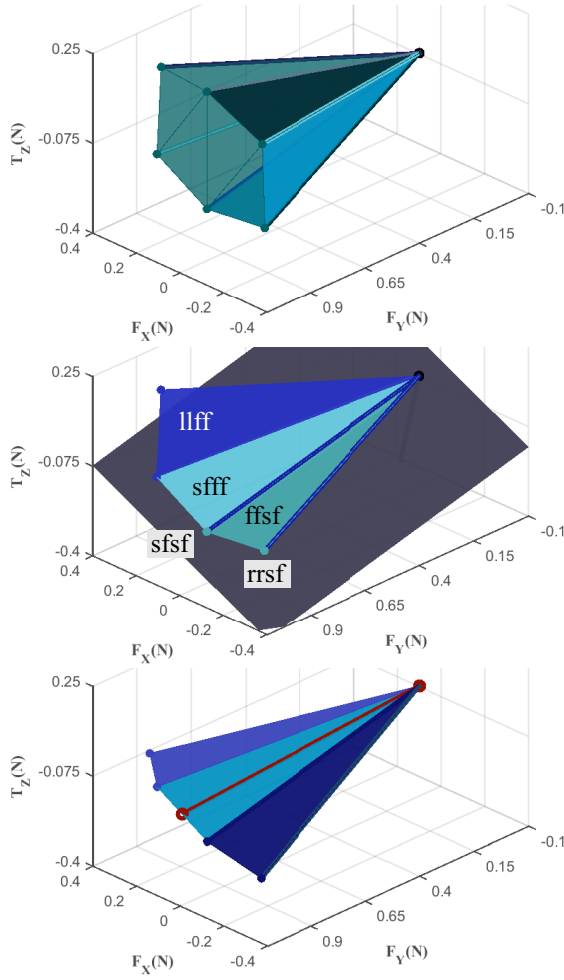


Fig. 3: Procedures of wrench stamping for the problem in Figure 1. The torque is scaled into Newton by the object length. Top: All nonempty cones. Middle: the cones of V-feasible modes (one or two generators). The gray plane is the force-controlled subspace. Bottom: Projection of feasible cones onto the force-controlled subspace. The red ray is the chosen wrench for mode “sfff”.

are PCCs in a lower dimensional space. Then we pick the force control η_{af} to be within the projection of the desired mode, while staying away from any other projections:

$$\eta_{af} \in \bar{\mathbb{C}}_{\text{goal}} - \bar{\mathbb{C}}_{\text{others}}. \quad (14)$$

The minus sign means set difference. If η_{af} exists, the actual w has to be in the desired mode because projections are surjective. An example of is shown in Figure 3, bottom. Because of the projection, we call this approach *Wrench Stamping*.

If the set $\bar{\mathbb{C}}_{\text{goal}} - \bar{\mathbb{C}}_{\text{others}}$ is empty, we call this mode *F-indistinguishable* since there is no force action the robot can take to avoid other feasible modes.

IV. ROBUSTNESS OF A CONTACT MODE

In this section, we show how to evaluate the robustness of a contact mode by looking at two causes of mode transitions:

A. Perturbations on Contact Geometry

Contact geometries plus friction coefficients determine the shape of hand and environment cones $\mathbb{C}_e, \mathbb{C}_h$. Modeling uncertainties change the shape of these cones, which could cause their intersection to disappear and the corresponding mode to be F-infeasible. For planar systems, we define a stability margin $\Phi_g(\mathbb{C}_e, \mathbb{C}_h)$ that describes the “depth” of the intersection, i.e. the minimal angular rotation $\mathbb{C}_e$ needs to take to eliminate non-trivial intersections with $\mathbb{C}_h$. Note that the volume (solid angle) of the intersection $\mathbb{C}_m$ is NOT the right metric. Cones of most modes have zero volume. To introduce our metric, we first define a distance function $\Delta(F, E)$ which computes the angular distance between ray E and hyperplane F :

$$\Delta(F, E) = \arcsin(n_F \cdot E), \quad (15)$$

where n_F denotes the normal of the hyperplane F . Both n_F and E are unit vectors. Next, define a function that evaluates the “depth” of a cone inside another cone. Consider two PCCs $\mathbb{C}_a, \mathbb{C}_b$ that satisfies $\mathbb{C}_b \subseteq \mathbb{C}_a$. Define

$$\sigma(\mathbb{C}_a, \mathbb{C}_b) = \min_j \max_i \Delta(F_j, E_i), \quad (16)$$

where F_j is the j th facet of $\mathbb{C}_a$, E_i is the i th edge of $\mathbb{C}_b$. A facet F_j defines a hyperplane with the positive normal pointing inside of the cone. We reduce dimension whenever possible, for example, if $\mathbb{C}_a$ has two generators, a facet only has one ray and defines a one-dimensional hyperplane. If $\mathbb{C}_a$ only has one generator, we define $\sigma(\mathbb{C}_a, \mathbb{C}_b) = 0$. Finally, define the *geometrical stability margin* $\Phi_g(\mathbb{C}_e, \mathbb{C}_h)$ as:

$$\Phi_g(\mathbb{C}_e, \mathbb{C}_h) = \min(\sigma(\mathbb{C}_e, \mathbb{C}_e \cap \mathbb{C}_h), \sigma(\mathbb{C}_h, \mathbb{C}_e \cap \mathbb{C}_h)). \quad (17)$$

B. Perturbations on Force Control

Disturbance forces can move the force control η_{af} out of the region defined by equation (14). We define the *control stability margin* Φ_c as the angular distance from η_{af} to the closest cone projections of other modes $\bar{\mathbb{C}}_{\text{others}}$. Figure 4 shows how the control in Figure 3 can fail under disturbance force.

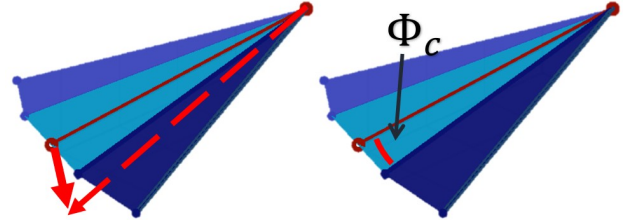


Fig. 4: Left: A disturbance force (bold red arrow) changes the contact mode. Right: illustration of the control stability margin.

Φ_c and Φ_g each describes the minimal disturbance force required to trigger mode transition in terms of angular distance in object wrench space. Define the *stability margin* Ψ :

$$\Psi = \min(\Phi_c, \Phi_g). \quad (18)$$

With proper scaling between force and torque, as well as a nominal force magnitude K_F , the product $K_F \Psi$ represents the minimal magnitude of the disturbance force that can change contact mode m . Just like grasping, we can make the system more robust simply by applying larger force.

We summarize wrench stamping in Procedure WrenchStamping. Line 2 of WrenchStamping calls 2D contact mode enumeration algorithm implemented in [16]. Line 1 uses the first half of the hybrid servoing algorithm [14].

Procedure WrenchStamping(Mode m^*)

Input: $\mathbf{N}, \mathbf{M}, \mathbb{C}_e, \mathbb{C}_h$ for all-fixed mode (9)(4)
Input: G, b_G , set of modes $\mathbb{M}$
Output: Ψ , HFVC ($n_{af}, n_{av}, \mathbf{R}_a, w_{av}, \eta_{af}$)

- 1 $[n_{af}, n_{av}, \mathbf{R}_a, w_{av}] = \text{hybridServoing}(\mathbf{N}_{m^*}, \mathbf{G}, b_G)$;
- 2 **if** *Crashing happens* ((12)(11)) **then return** $\emptyset$;
- 3 Initialize empty set $\hat{\mathbb{M}} \leftarrow \emptyset$;
// Mode filtering by F-feasibility
- 4 **foreach** $m \in \mathbb{M}$ **do**
5 Assemble $\mathbb{C}_e^{(m)}, \mathbb{C}_h^{(m)}$ from columns of $\mathbb{C}_e, \mathbb{C}_h$;
6 **if** $\mathbb{C}_e^{(m)} \cap \mathbb{C}_h^{(m)} = \emptyset$ **then Continue**;
7 Compute stability margin $\Phi_g^{(m)}(\mathbb{C}_e^{(m)}, \mathbb{C}_h^{(m)})$ (17);
8 **if** $\Phi_g^{(m)} = 0$ **then Continue**;
9 Assemble $\mathbf{N}_m, \mathbf{M}_m$ from rows of $\mathbf{N}, \mathbf{M}$;
10 $\hat{\mathbb{M}} \leftarrow \{\hat{\mathbb{M}}, m\}$;
- 11 **end**
- 12 $\mathbb{M} \leftarrow \hat{\mathbb{M}}, \hat{\mathbb{M}} \leftarrow \emptyset$;
// Mode filtering by V-feasibility
- 13 **foreach** $m \in \mathbb{M}, m \neq m^*$ **do**
14 **if** (7)(9) *are feasible* **then** $\hat{\mathbb{M}} \leftarrow \{\hat{\mathbb{M}}, m\}$;
- 15 **end**
// Projections
- 16 $\bar{\mathbb{C}}_{\text{goal}} \leftarrow \mathbf{R}_{af}^{-T} \mathbb{C}_m, \mathbf{R}_{af}^{-1}$ is the first n_{af} cols of $\mathbf{R}_a^{-1}$;
- 17 $\bar{\mathbb{C}}_{\text{others}} \leftarrow \emptyset$;
- 18 **foreach** $m \in \hat{\mathbb{M}}$ **do**
19 $\bar{\mathbb{C}}_{\text{others}} \leftarrow \{\bar{\mathbb{C}}_{\text{others}}, \mathbf{R}_{af}^{-T} \mathbb{C}_m\}$
- 20 **end**
- 21 $\bar{\mathbb{C}}_{\text{remain}} \leftarrow \bar{\mathbb{C}}_{\text{goal}} - \bar{\mathbb{C}}_{\text{others}}$;
- 22 **if** $\bar{\mathbb{C}}_{\text{remain}} = \emptyset$ **then return** $\emptyset$;
- 23 Pick $\eta_{af} \in \bar{\mathbb{C}}_{\text{remain}}$, compute Φ_c ;
- 24 $\Psi \leftarrow \min(\Phi_c, \Phi_g)$ (Φ_g was computed in line 7);

V. ALGORITHMS FOR TWO PROBLEMS

In this section, we provide algorithms for two problems in shared grasping.

A. Shared Grasping Control with Mode Selection

We provide algorithm 1 to solve robust control for shared grasping problem. Given a desired velocity, the algorithm finds the most robust contact mode and computes a hybrid force-velocity control.

If a desired contact mode is given, the control problem becomes a hybrid servoing problem [14]. Procedure WrenchStamping itself can solve it better than the original hybrid servoing algorithm: users do not need to provide the guard conditions, WrenchStamping determines the condition of mode transition by itself. Also WrenchStamping can handle robust control under sliding contacts.

Algorithm 1: Shared Grasping Control

Input: $\mathbf{N}, \mathbf{M}, \mathbb{C}_e, \mathbb{C}_h$ for all-fixed mode (9)(4), G, b_G

Output: The chosen contact mode m , stability margin Ψ , HFVC ($n_{af}, n_{av}, \mathbf{R}_a, w_{av}, \eta_{af}$)

- 1 Compute the cone of all-fixed mode: $\mathbb{C}_{AF} \leftarrow \mathbb{C}_e \cap \mathbb{C}_h$;
- 2 $\mathbb{M}_{\text{all}} \leftarrow$ all contact modes, $\mathbb{M} \leftarrow \emptyset$;
- 3 Line 4 - 11 of Procedure WrenchStamping for $m \in \mathbb{M}_{\text{all}}$, save results in $\mathbb{M}$;
- 4 Initialize the solution set: $\mathbf{S} \leftarrow \emptyset$;
- 5 **foreach** $m \in \mathbb{M}$ **do**
6 $\mathbf{S} \leftarrow \{\mathbf{S}, \text{WrenchStamping}(m)\}$
- 7 **end**
- 8 **return** the $s \in \mathbf{S}$ with highest stability margin Ψ

B. Geometry Optimization

We can optimize contact geometry parameters to maximize the geometrical stability margin Φ_g for a given mode. Note that the function $\Phi_g(\mathbb{C}_e, \mathbb{C}_h)$ boils down to the Δ distance between a facet F and an edge E . The facet has either two generators C_i, C_j or just C_i , we can backtrack its location in $\mathbb{C}_e$ and $\mathbb{C}_h$. The edge is a generator of $\mathbb{C}_e \cap \mathbb{C}_h$, which could be either an edge C_k of $\mathbb{C}_e$ or $\mathbb{C}_h$, or an intersection of a facet (C_1, C_2) in $\mathbb{C}_e$ and a facet (C_3, C_4) in $\mathbb{C}_h$. In the latter case, the expression of the edge is $(C_1 \times C_2) \times (C_3 \times C_4)$. So we can explicitly write down Φ_g as a function of contact screws:

$$\Phi_g(\mathbb{C}_e, \mathbb{C}_h) = \arccos \left(\frac{C_i}{\|C_i\|} \cdot \frac{C_k}{\|C_k\|} \right) \quad \text{or} \quad (19)$$

$$= \arccos \left(\frac{C_i}{\|C_i\|} \cdot \frac{(C_1 \times C_2) \times (C_3 \times C_4)}{\|(C_1 \times C_2) \times (C_3 \times C_4)\|} \right).$$

for one-dimensional facet F , or

$$\Phi_g(\mathbb{C}_e, \mathbb{C}_h) = \arcsin \left(\frac{C_i \times C_j}{\|C_i \times C_j\|} \cdot \frac{C_k}{\|C_k\|} \right) \quad \text{or} \quad (20)$$

$$= \arcsin \left(\frac{C_i \times C_j}{\|C_i \times C_j\|} \cdot \frac{(C_1 \times C_2) \times (C_3 \times C_4)}{\|(C_1 \times C_2) \times (C_3 \times C_4)\|} \right)$$

for two-dimensional F . We can further expand the contact screws into contact geometry parameters and compute the

gradient of Φ_g in analytical form. We outline a gradient descent algorithm in Algorithm 2. The parameter $\mathbf{p}$ may contain contact locations, normals and friction coefficients. Note that computation in (19) and (20) only involves contact screws that are the current bottleneck of stability margin. This set of contact screws will change between iterations. We observed this phenomenon in the experiments section.

In order to get analytical expression of the gradient of Φ_g , we ignore the normalizations in (19) and (20) when computing their derivatives. To obtain the correct gradient about a contact screw, we just project the gradient computed this way onto the tangent plane of the contact screw, as shown in line 7.

Algorithm 2: Geometrical Parameter Optimization

Input: Contact geometry parameter $\mathbf{p}$

Output: Optimized contact geometry parameter $\mathbf{p}^*$

```

1 repeat
  // 1. Evaluate  $\Phi_g$ 
2  Compute  $\mathbb{C}_e, \mathbb{C}_h$  (4) using  $\mathbf{p}$ ;
3  Compute  $\mathbb{C}_m = \mathbb{C}_e \cap \mathbb{C}_h$ , record the mapping  $M_1$ 
   between columns of  $\mathbb{C}_m$  and columns of  $\mathbb{C}_e, \mathbb{C}_h$ ;
4  Compute  $\Phi_g$  (17), record the mapping  $M_2$  between
    $C_{i,j,k,1,2,3,4}$  and columns of  $\mathbb{C}_e, \mathbb{C}_h, \mathbb{C}_m$ ;
5  Use  $M_1$  and  $M_2$  to compute the mapping  $M_3$ 
   between  $C_{i,j,k,1,2,3,4}$  and columns of  $\mathbb{C}_e, \mathbb{C}_h$ ;
  // 2. Compute gradient
6  Compute  $\Phi_t = \partial\Phi_g/\partial C_t, t \in \{i, j, k, 1, 2, 3, 4\}$  by
   differentiating (19)(20) without denominators;
7  Remove the components of  $\Phi_t$  that scales  $C_t$ :
    $\Phi_t = \Phi_t - \Phi_t \cdot C_t, t \in \{i, j, k, 1, 2, 3, 4\}$ ;
8  Use  $M_4$  to construct  $\Phi_C = \partial\Phi_g/\partial C$  from  $\Phi_t, C \in$ 
   columns of  $\mathbb{C}_e, \mathbb{C}_h$ ;
9  Compute the Jacobian  $C_p = \partial C/\partial \mathbf{p}$  by
   differentiating the expression of  $\mathbb{C}_e, \mathbb{C}_h$  (4);
10  $\Phi_p \leftarrow \sum_C (\Phi_C \cdot C_p)$ ;
   // 3. Update
11  $\mathbf{p} \leftarrow \mathbf{p} + d\Phi_p, d$  is a step length;
12 until converge;
```

VI. EXPERIMENTS

We implement our algorithms in MATLAB and test them on several tasks in experiments. We use an ABB IRB120 robot in section VI-A and VI-B, a UR5e robot in section VI-D, both are 6-axis robot and position-controlled; we implement a hybrid force-velocity controller with a wrist-mounted ATI Mini-40 FT sensor. The communication rate is 250Hz for the ABB robot and 500Hz for the UR5e robot. In both cases, the force control bandwidth is roughly 1Hz.

A. Robust Control of Desired Contact Modes

First, we validate the ability of our method in computing robust actions under different contact modes. Consider the shared grasping system in Figure 1 as an example. The friction

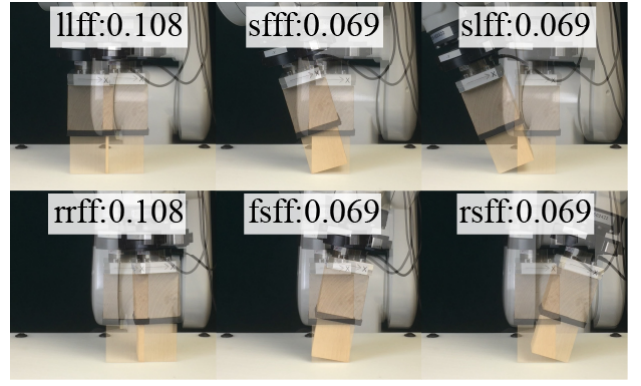


Fig. 5: Different contact modes and their stability margins. The order of contacts follows Figure 1.

coefficients are estimated to be 1.2 and 0.25 for hand-object and table-object contacts, respectively.

There are 81 contact modes for the four contact points. Algorithm 1 computes 13 modes with positive geometrical stability margin Φ_g . We hand-pick six modes with object motion and solve them with Procedure WrenchStamping. The results are shown in Figure 5.

B. Contact Mode Selection and Control

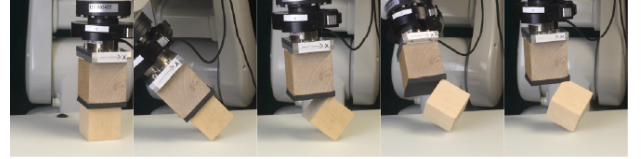


Fig. 6: Flipping a cube on its corners.

Using the same object and robot, we demonstrate Algorithm 1 in the task of cube rotation with a palm, as shown in Figure 6. The task is challenging because the object could fall at any time, the control must always be robust. The task has four stages. The goal velocity for the first two stages are:

$$\mathbf{G}_1 = [0 \ 0 \ 1 \ 0 \ 0 \ 0], \quad b_{G1} = -0.1. \quad (21)$$

$$\mathbf{G}_2 = \begin{bmatrix} Adj_{WH} & \mathbf{I}_3 \\ 0 \ 0 \ 0 & 0 \ 0 \ 1 \end{bmatrix}, \quad b_{G2} = [0 \ 0 \ 0 \ -0.1]^T. \quad (22)$$

First three rows of $\mathbf{G}_2, b_{G2}$ constrain the object to be static. Stage three and four are the same but in a different plane. Each stage is divided into eight time steps, at which Algorithm 1 decides which mode and what action to take. There is no vision feedback, the object pose is estimated from robot hand pose and has an accumulating error. We execute the four-stage task eight times and have four complete successes. The failures all happen in the last two stages.

C. Finger Placement Optimization

We test Algorithm 2 on a point finger manipulation task, as shown in Figure 7, left. The width and height of the

block are both 0.1m. We optimize finger location for two tasks: 1. pivoting about the left corner with mode “sff”; 2. Sliding to the left with mode “llf”. We show the result of 200 iterations of gradient descent in Figure 8, right, which takes 0.45s in MATLAB. The stability margin increases and reaches a plateau. Note that on the plateau the solution is still zig-zagging by a small magnitude, because the set of contact screws being optimized is varying between iterations, as explained in section V-B.

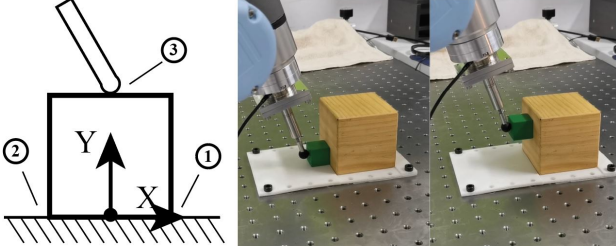


Fig. 7: Left: the Finger-block example and the ordering of contacts. Right: Experiment setup. The yellow block is fixed.

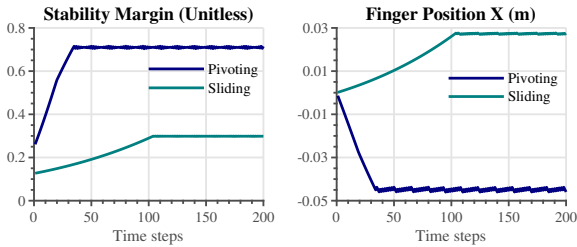


Fig. 8: The evolution of finger position and stability margin during the optimization.

D. Compute the Range of Feasible Parameters

Consider again the task of block sliding (Figure 7). Using the optimized finger location and a HFVC computed from Algorithm 1, we compute the range of table contact locations that produce a positive stability margin for the desired mode “llf”. We randomly sample 1000 pairs of object right and left corner locations within the range $(-0.2m, 0.2m)$. The result is shown in Figure 9. A point is marked green if the stability margin is positive. We can see there is roughly a safety range of $0.05m$ around the nominal point, i.e. the sliding motion can still be successful if the actual environmental contact location changes by $0.05m$. We can do the same computation for varying object height, friction coefficient, etc.

We implement this sliding action with a wide range of objects and also with gravity as disturbance force. The results are shown in Figure 10.

VII. CONCLUSION AND DISCUSSION

To conclude, we provide stability analysis method for shared grasping problems that involves multiple contact modes. We demonstrate how to find the most robust contact mode and compute robot actions to achieve the desired system velocity,

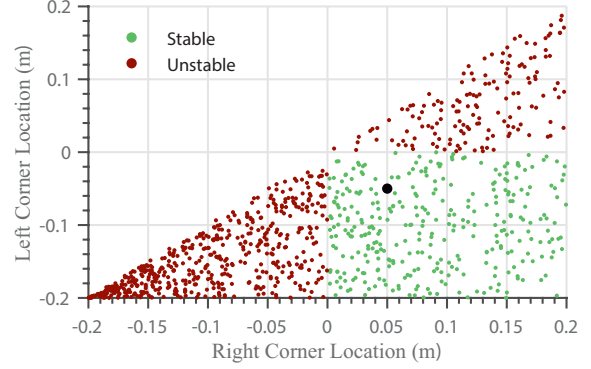


Fig. 9: Feasibility of different model parameters. The control is computed for the black dot.

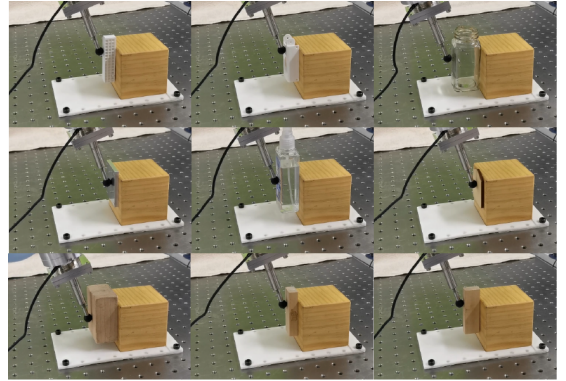


Fig. 10: The same control law works for different objects.

as well as how to refine finger placements to improve stability. We are working on generalizing the method to 3D problems. We need to revisit the design of stability margins ((17) and (18)), which currently relies on 3D geometry.

We demonstrate the robustness of our method in a variety of experiments. The speed of robot motion is limited by the bandwidth of the robot. Recent progress on direct-drive robot hands [3] opens possibilities for faster shared grasping.

Although shared grasping is designed for making environmental contacts, the stability analysis and algorithms also work for real in-hand manipulation problems if there are two (groups of) fingers.

Shared grasping complements grasping as another category of manipulations with good robustness. This work shows its potential for contact-rich manipulations that are reliable enough for real-world applications.

ACKNOWLEDGMENTS

We would like to thank Eric Huang and Xianyi Cheng for insightful discussions. We also want to thank Nikhil Chavan-Dafle and Ankit Bhatia for their suggestions in the writing of the paper.

REFERENCES

- [1] Yasumichi Aiyama, Masayuki Inaba, and Hirochika Inoue. Pivoting: A new method of graspless manipulation

- of object by robot fingers. In *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, volume 1, pages 136–143. IEEE, 1993.
- [2] Dmitry Berenson, Siddhartha S Srinivasa, Dave Ferguson, and James J Kuffner. Manipulation planning on constraint manifolds. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 625–632. IEEE, 2009.
 - [3] Ankit Bhatia, Aaron M. Johnson, and Matthew T. Mason. Direct drive hands: Force-motion transparency in gripper design. In *Proceedings of Robotics: Science and Systems*, June 2019.
 - [4] Nikhil Chavan-Dafle and Alberto Rodriguez. Prehensile pushing: In-hand manipulation with push-primitives. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 6215–6222, 2015.
 - [5] Nikhil Chavan-Dafle and Alberto Rodriguez. Sampling-based planning of in-hand manipulation with external pushes. *arXiv preprint arXiv:1707.00318*, 2017.
 - [6] Nikhil Chavan-Dafle, Rachel Holladay, and Alberto Rodriguez. In-hand manipulation via motion cones. *Robotics: Science and Systems*, 2018.
 - [7] Nikhil Chavan Dafle, Alberto Rodriguez, Robert Paolini, Bowei Tang, Siddhartha S Srinivasa, Michael Erdmann, Matthew T Mason, Ivan Lundberg, Harald Staab, and Thomas Fuhlbrigge. Extrinsic dexterity: In-hand manipulation with external forces. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1578–1585. IEEE, 2014.
 - [8] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017.
 - [9] Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3389–3396. IEEE, 2017.
 - [10] Weiqiao Han and Russ Tedrake. Local trajectory stabilization for dexterous manipulation via piecewise affine approximations. *arXiv preprint arXiv:1909.08045*, 2019.
 - [11] François Robert Hogan and Alberto Rodriguez. Feedback control of the pusher-slider system: A story of hybrid and underactuated contact dynamics. *arXiv preprint arXiv:1611.08268*, 2016.
 - [12] Anne Holladay, Robert Paolini, and Matthew T Mason. A general framework for open-loop pivoting. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3675–3681, 2015.
 - [13] Yifan Hou and Matthew T Mason. Criteria for maintaining desired contacts for quasi-static systems. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, November 2019.
 - [14] Yifan Hou and Matthew T Mason. Robust execution of contact-rich motion plans by hybrid force-velocity control. In *International Conference on Robotics and Automation (ICRA) 2019*. IEEE Robotics and Automation Society (RAS), May 2019.
 - [15] Yifan Hou, Zhenzhong Jia, and Matthew T Mason. Fast planning for 3d any-pose-reorienting using pivoting. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1631–1638. IEEE, 2018.
 - [16] Eric Huang, Xianyi Cheng, and Matthew T Mason. Efficient contact mode enumeration in 3d. In *Workshop on the Algorithmic Foundations of Robotics*, 2020.
 - [17] Imin Kao and Mark R Cutkosky. Quasistatic manipulation with compliance and sliding. *The International journal of robotics research*, 11(1):20–40, 1992.
 - [18] Michael Koval. *Robust Manipulation via Contact Sensing*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, September 2016.
 - [19] Kevin M Lynch and Matthew T Mason. Stable pushing: Mechanics, controllability, and planning. *The International Journal of Robotics Research*, 15(6):533–556, 1996.
 - [20] Matthew T Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6):418–432, 1981.
 - [21] Matthew T Mason. *Mechanics of robotic manipulation*. MIT press, 2001.
 - [22] N Harris McClamroch and Danwei Wang. Feedback stabilization and tracking of constrained robots. *IEEE Transactions on Automatic Control*, 33(5):419–426, 1988.
 - [23] Igor Mordatch, Zoran Popović, and Emanuel Todorov. Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, pages 137–144. Eurographics Association, 2012.
 - [24] Igor Mordatch, Emanuel Todorov, and Zoran Popović. Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics (TOG)*, 31(4):43, 2012.
 - [25] Igor Mordatch, Kendall Lowrey, and Emanuel Todorov. Ensemble-cio: Full-body dynamic motion planning that transfers to physical humanoids. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 5307–5314. IEEE, 2015.
 - [26] Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.
 - [27] Michael Posa, Cecilia Cantu, and Russ Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research*, 33(1):69–81, 2014.
 - [28] Marc H Raibert and John J Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103(2):126–133, 1981.
 - [29] J Kenneth Salisbury. Whole-arm manipulation. In *Proc. of the 4th Int. Symp. of Robotics Research*. Published by the MIT Press, 1987.

- [30] Naoyuki Sawasaki, Masayuki Inaba, and Hirochika Inoue. Tumbling objects using a multi-fingered robot. In *Proceedings of the 20th International Symposium on Industrial Robots and Robot Exhibition*, pages 609–616, 1989.
- [31] Arne Sieverling, Clemens Eppner, Felix Wolff, and Oliver Brock. Interleaving motion in contact and in free space for planning under uncertainty. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4011–4073. IEEE, 2017.
- [32] Raúl Suárez, Jordi Cornella, and Máximo Roa Garzón. *Grasp quality measures*. Citeseer, 2006.
- [33] Hajime Terasaki and Tsutomu Hasegawa. Motion planning of intelligent manipulation by a parallel two-fingered gripper equipped with a simple rotating mechanism. *IEEE Transactions on Robotics and Automation*, 14(2):207–219, 1998.
- [34] Marc Toussaint, Kelsey Allen, Kevin A Smith, and Joshua B Tenenbaum. Differentiable physics and stable modes for tool-use and manipulation planning. In *Robotics: Science and Systems*, 2018.
- [35] Masaru Uchiyama and Pierre Dauchez. A symmetric hybrid position/force control scheme for the coordination of two robots. In *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pages 350–356. IEEE, 1988.
- [36] Dustin J Webb and Jur Van Den Berg. Kinodynamic rrt*: Asymptotically optimal motion planning for robots with linear dynamics. In *2013 IEEE International Conference on Robotics and Automation*, pages 5054–5061. IEEE, 2013.
- [37] Harry West and Haruhiko Asada. A method for the design of hybrid position/force controllers for manipulators constrained by contact with the environment. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 251–259. IEEE, 1985.
- [38] Eiichi Yoshida, Mathieu Poirier, Jean-Paul Laumond, Oussama Kanoun, Florent Lamiraux, Rachid Alami, and Kazuhito Yokoi. Pivoting based manipulation by a humanoid robot. *Autonomous Robots*, 28(1):77–88, 2010.
- [39] Tsuneo Yoshikawa. Dynamic hybrid position/force control of robot manipulators—description of hand constraints and calculation of joint driving force. *IEEE Journal on Robotics and Automation*, 3(5):386–392, 1987.
- [40] Andy Zeng, Shuran Song, Stefan Welker, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. 2018.
- [41] Jiaji Zhou, Robert Paolini, J Andrew Bagnell, and Matthew T Mason. A convex polynomial force-motion model for planar sliding: Identification and application. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 372–377. IEEE, 2016.
- [42] Jiaji Zhou, Robert Paolini, Aaron M Johnson, J Andrew Bagnell, and Matthew T Mason. A probabilistic planning framework for planar grasping under uncertainty. *IEEE Robotics and Automation Letters*, 2(4):2111–2118, 2017.