Criteria for Maintaining Desired Contacts for Quasi-Static Systems

Yifan Hou and Matthew T. Mason *Fellow, IEEE*

***Abstract*— In robotic manipulation, finding a feasible motion plan doesn't guarantee a successful execution. The real world could bring all kinds of unexpected changes to the planned motion, the most deadly ones are usually marked by or caused by unexpected changes of contacts (object slipping away between fingers; getting stuck somewhere, etc). We notice that some actions are more likely to maintain desired contacts than others. To help finding these actions, in this work we propose a set of criteria to quantify the robustness of contacts against modeling uncertainties and disturbance forces. Under the quasi-static assumption, we analyze the causes of contact mode (sticking, sliding, disengaged) transitions and discuss how to endure larger uncertainties and disturbances. We summarize our results into several physically meaningful and easy-to-compute scores, which can be used to evaluate the quality of each individual contacts in a manipulation system. We illustrate the meaning of the scores with a simple example.**

I. Introduction

A contact could have one of three modes: sticking, sliding or disengaged. Robot's failures in manipulation are usually associated with unexpected modes of some contacts. To successfully execute a task, it is important to maintain desired contact modes under real world uncertainties and disturbances. Many factors can influence the robustness of maintaining a contact mode against modeling uncertainties and disturbance forces, such as the choice of contact locations, robot forces and/or robot velocities.

For simple systems, people can tell the difference from their physical intuition. In the example shown in Fig. 1 (a) and (b), a robot is pushing a block to the left with one finger (the black stick). The robot actions include force and velocity controls in different directions. Comparing the two actions shown in (a) and (b), you can probably tell that the motion in (a) is less likely to get stuck.

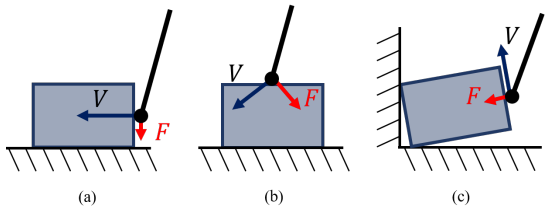

Fig. 1. (a)(b): different robot actions for the block pushing task. (c) levering up a block against a corner. In all figures, the black stick denotes the robot finger.

For more complicated systems, it is no longer intuitive to judge how good a control is. For example, in Fig. 1c, the robot finger is levering up a block against a corner. Given a certain robot action, we can ask the questions: how reliable can the robot maintain the sliding contacts between the object and the walls? how reliable can the robot maintain the sticking contact between the object and the finger?

In this work, we try to answer such questions by evaluating the amount of modeling uncertainties and external disturbance forces required to break a given contact mode. To be general, we consider the case when the robot can do both force control and velocity control (hybrid force-velocity control, which includes the case of pure velocity or force control). We summarize the results into a set of scores to quantify the robustness of each individual contact for a general manipulation system.

There are three possible modes for a contact: sticking, sliding, or disengaged. The maintaining of the last one belongs to the task of collision avoidance and is not covered in this paper. We focus on how to make sure the already made contacts stay in their desired modes.

We make the following assumptions throughout the paper:

- Motions are quasi-static, i.e. inertial forces are negligible.
- Velocity control can generate arbitrarily large forces when necessary.
- We consider rigid body contact model with Coulomb friction plus stiction. The magnitude of stiction is the same as sliding friction.
- Maximum-dissipation principle holds, i.e. sliding friction force of a point contact is in the opposing direction of the sliding velocity.

Any model parameter may have uncertainties, including but not limited to the object shape and friction coefficients.

Our analysis works for point-to-face contacts, which can be generalized to any rigid body contact type that has a clear definition of contact normal direction. Such generalization usually brings redundant constraints and contact force variables; we will discuss the influence of redundancy in Section IV-A.

II. Related Work

A. Maintaining Contact Modes

Roboticists have managed to utilize contacts in many specific scenarios. For example, a parallel gripper can alternate between sliding and sticking contacts by regulating the gripping force [3], [7], [22]. When contact normal force was limited, people used large external forces [20] or velocity-controlled external contacts [1] to ensure sliding. Pushing is a good example where the sliding and sticking of the

*This work was supported under NSF Grant No. 1662682.

The authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA. yifanh@cmu.edu, matt.mason@cs.cmu.edu

finger on the object can be controlled by the direction of pusher velocity with respect to the motion cone [12], [6], [2]. Our work generalizes the analysis to three dimensional pushing with potentially more contacts with the environment, in which case the robot action may cause the system to crash under huge forces.

Closely related to maintaining contact modes, the concepts *jamming* and *wedging* refer to two failure cases in mechanical assembly, which was analyzed in detail by Whitney [24]. Roughly speaking, jamming means a sliding contact turns into sticking unexpectedly under low-stiffness control. Wedging means two point contacts between an object and the environment form a force closure and fully immobilize the object. Conditions for checking jamming and wedging are available in [5], [21]. Maintaining sliding contact with low-stiffness control is vulnerable in many ways. In this work we use velocity control instead to avoid jamming completely. We only need to consider avoiding crashing (Section IV-A), which is easier to analyze. We also don't need to worry about wedging, because the task of maintaining contact modes already avoids it.

Another example of maintaining contacts is in legged locomotion. The most widely used condition for walking stability is the Zero Moment Point (ZMP) [23] criterion. By ensuring the ZMP stays within the legged robot's support-polygon, the robot's supporting foot shall remain in a sticking face-to-face contact with the ground. ZMP assumes high friction (no slipping) and boils down to maintaining positive contact forces for a dynamic multi-link system with no other external contacts.

B. Dynamic Control of Systems with Contacts

The dynamic locomotion community is concerned with the stability of a legged robot walking on the ground [17], which is related to manipulation through contacts. Methods are available for evaluating the robustness of a walking robot under discontinuous impact from the ground [18]. However, it's tricky to apply them to manipulation problems, because these work usually does not consider crashing avoidance and the possibility of sliding contacts.

C. Robust Planning

Most work on motion planning, especially planning through contacts, is focused on feasibility instead of robustness [16], [14]. There are a few exceptions. Stable pushing [12] ensures a sticking contact between the pusher and the object by pushing in suitable directions; Push-grasping [4] maximizes the probability of successful grasps by pushing towards the most probable region; Convergence planning [10] utilizes convergence analysis for continuous (or close to continuous) vector fields to compute a trajectory with a converging funnel.

D. Hybrid Servoing

To handle contacts, it's not enough to just move exactly in the feasible direction computed from the contact Jacobians [15] since the modeling may not be accurate. To further handle modeling uncertainties, people introduced mechanical compliance [9] and active compliance [19] to avoid excessive internal forces. When combining accurate velocity control and compliant force control together, it is possible to avoid crushing and maintain accuracy simultaneously [13], [15]. In [8], we proposed hybrid servoing to compute robust hybrid force-velocity controls automatically given a motion plan. However, the criteria being optimized in [8] was designed by physical intuitions. In this work we show that the cost function is an approximation to one of the robustness scores proposed in this paper. The scores can be used to build a better hybrid servoing algorithm.

We organize the paper as follows. In Section III we introduce the background of the quasi-static modeling method, focusing on the implications that are crucial for our analysis. We then introduce a set of robustness criteria with clear physical interpretations in Section IV. Finally in Section V we exemplify the scores under different actions for several simple manipulation systems.

III. Quasi-Static Modeling

Consider a system of rigid bodies that contains n_a actuated Degree-Of-Freedoms (DOF) and n_u unactuated DOF. For example, for a system of one robot arm and one object, the actuated DOF are the robot joints, while the unactuated DOF is the 6-dimensional object pose. Denote $n = n_a + n_u$ as the total DOF of the system. Denote $v, f \in \mathbb{R}^n$ as the generalized velocity/force vectors of the system.

A. Describe the Contacts

Consider a system of l point-to-face contacts. Other types of contacts can either be approximated by multiple point contacts [1], or be analyzed similarly. Denote the normal distance and tangent displacements for contact i as $\Phi_n^{(i)}$ and $\Phi_t^{(i)}, \Phi_o^{(i)}$, respectively. Denote their Jacobians about the system configuration as $J_n^{(i)}$, $J_t^{(i)}$, and $J_o^{(i)}$. Denote Ω as the mapping from the generalized velocity to the time derivative of the system configuration. Then we can use

$$M^{(i)} := \begin{bmatrix} J_n^{(i)} \\ J_t^{(i)} \\ J_o^{(i)} \end{bmatrix} \Omega \tag{1}$$

to map between the local contact force $\lambda^{(i)}$/velocity $v_L^{(i)}$ and the generalized force/velocity:

$$\begin{aligned} v_L^{(i)} &= M^{(i)} v, \\ f_i &= M^{(i)T} \lambda^{(i)}. \end{aligned} \tag{2}$$

Here f_i is the contribution of contact force i to the generalized force vector. Denote M, λ as the vertical concatenations of all $M^{(i)}$ and $\lambda^{(i)}$.

Contacts also introduces the holonomic constraint on the generalized velocity v. This constraint only comes from the contact normal force and the sticking friction. To properly describe this constraint, define

$$N^{(i)} := \begin{cases} J_n^{(i)}\Omega & \text{if contact } i \text{ is sliding} \\ M^{(i)} & \text{if contact } i \text{ is sticking} \end{cases} \tag{3}$$

Then the holonomic constraint is

$$Nv = 0. \tag{4}$$

where $N = \begin{bmatrix} N^{(1)} \\ \vdots \\ N^{(l)} \end{bmatrix}$. We assume the natural constraint itself is feasible, otherwise the manipulation problem is meaningless.

Contacts also introduce several constraints on the local contact forces:

$$A_z \lambda \geq 0, \tag{5}$$

$$A_{\text{cone}} \lambda \leq b_{\text{cone}}, \tag{6}$$

$$A_{\text{slide}} \lambda = b_{\text{slide}}. \tag{7}$$

(5) describes the minimal normal force constraint. Matrix A_z select the contact normal forces from the contact force vector λ. (6) describes the friction cone constraint. (7) describes the sliding friction constraint.

B. Newton's Second Law

Under quasi-static assumption, the forces in the system are always in static equilibrium, i.e. inertia force does not exist in the Newton's Second Law:

$$M^T \lambda + f + F = 0, \tag{8}$$

where F is the external force vector (such as gravity).

Unlike dynamic systems which are driven by force, quasi-static systems ignore impacts and inertia forces, thus break the force-motion relation provided by Newton's Second Law. The benefit is to have a simpler system with fewer variables (no accelerations). A quasi-static system is driven by velocity constraints, which come from the environment and the robot action.

C. Describe Robot Actions

To be general, we assume the robot uses Hybrid Force-Velocity Control (HFVC) (also called hybrid force-position control). Pure force control and pure velocity control are special cases of HFVC. Without losing generality, assume the last n_a DOF of the generalized vectors belong to the robot action. A HFVC action can be described by the following variables:

- $n_{av}, n_{af} \in \mathbb{R}$: DOF of velocity/force command, respectively. $n_{av} + n_{af} = n_a$.
- $C \in \mathbb{R}^{n_{av} \times n}, b_C \in \mathbb{R}^{n_{av}}$: velocity control:

$$Cv = b_C. \tag{9}$$

- $D \in \mathbb{R}^{n_{af} \times n}, b_D \in \mathbb{R}^{n_{af}}$: force control:

$$Df = b_D. \tag{10}$$

Matrices C and D describes the direction of velocity and force controlled axes. We assume their directions are perpendicular, because there is no benefit of making them not perpendicular, and it's always possible to choose the force controlled axes to be perpendicular to velocity controlled axes [8].

Before we proceed, let's clarify a few concepts about the velocity command. Each row of C specifies a *velocity-controlled direction*. There could be multiple velocity-controlled directions, the force in each of these directions could be arbitrarily large to overcome resistance and maintain the desired velocity specified by C and b_C. The least-square solution of (9), $v^* = C^\dagger b_C$, is called the *commanded velocity*. This is the actual velocity if there is no motion in the force controlled directions. A 2D example is shown in Fig. 2. An important observation is that, the commanded velocity

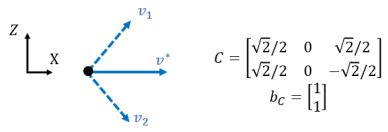

Fig. 2. Velocity command. The dashed lines are the velocity-controlled directions. The solid red line is the commanded velocity.

at a contact point *does not have to be within the contact tangent plane*. An example is shown in Fig. 3 The actual velocity, which must be along the contact tangent, is the combination of the velocity command and *the velocity in the force controlled directions*. Under quasi-static assumption,

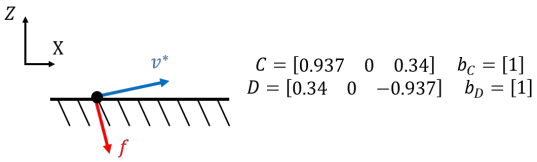

Fig. 3. Velocity and force command on a point. The velocity command alone does not need to align with the contact tangent plane.

the contact will stay engaged as long as there is a force component in the inward normal direction.

D. Problem Formulation

Given the system model (N, M, A_z, A_{cone}, b_{cone}, A_{slide}, b_{slide} and F) at a time step, evaluate the robustness of an action described by n_{av}, n_{af}, C, D, b_C and b_D in terms of maintaining current contact modes under modeling uncertainties and force disturbances.

IV. Failure Modes and Robustness Scores

In this section, we discuss all possible ways a contact mode can break, and provide scores to measure the robustness of the contacts against each situation.

As a preparation step, we firstly compute a set of nominal values for the variables v, f and λ given an action and the system model. Due to the flaw of rigid body modeling, there could be multiple solutions for contact forces. We only pick a unique one by minimizing the magnitude of f and λ.

A. Crashing

Crashing means the robot experiences huge internal force after a small motion. For example, the robot will crash if it tries to penetrate a rigid wall. Generally speaking, a robot

system crashes when the velocity constraints (9) and (4) become infeasible. Denote $\Lambda = \begin{bmatrix} C \\ N \end{bmatrix}$, $b_\Lambda = \begin{bmatrix} b_C \\ 0 \end{bmatrix}$, write down (9)(4) as

$$\Lambda v = b_\Lambda. \tag{11}$$

To avoid crashing, the linear system (11) must remain feasible under modeling uncertainties (force disturbances have no influence on (11)). The uncertainties are reflected as perturbations on the coefficients of (11):

$$(\Lambda + \Delta_\Lambda)\, v = b_\Lambda + \delta_b, \tag{12}$$

where Δ_Λ, δ_b are the unknown perturbations.

From linear algebra we know that if the rows of $\Lambda + \Delta_\Lambda$ are linearly independent, the pseudo-inverse $(\Lambda + \Delta_\Lambda)^\dagger$ will satisfy

$$(\Lambda + \Delta_\Lambda)(\Lambda + \Delta_\Lambda)^\dagger = I, \tag{13}$$

then the linear system (12) will always have a solution $(\Lambda + \Delta_\Lambda)^\dagger(b_\Lambda + \delta_b)$ regardless of δ_b. In other words, if $\Lambda + \Delta_\Lambda$ has independent rows, the robot will not crash. Row independence of $\Lambda + \Delta_\Lambda$ can be measured by its smallest singular value:

$$\sigma_{\min}(\Lambda + \Delta_\Lambda) > 0. \tag{14}$$

In order to satisfy (14) under unknown Δ_Λ, we need make the smallest singular value of Λ as big as possible by picking a suitable action C:

$$\max_C \sigma_{\min}(\Lambda) = \sigma_{\min}(\begin{bmatrix} C \\ N \end{bmatrix}). \tag{15}$$

The larger $\sigma_{\min}(\Lambda)$ is, the more modeling uncertainties the system can handle before crashing.

1) Redundancy in N: Sometimes the rows of the natural constraint N aren't linearly independent. This occurs when we have redundant contact modeling, for example when we use multiple point-to-face contacts to represent a face-to-face contact. In this case, Λ is always linearly dependent. The key of handling this situation is the fact that the natural constraint $Nv = 0$ still always has solution(s). Then the following is a row-independent equivalence to $Nv = 0$:

$$\mathrm{Row}(N)v = 0, \tag{16}$$

where the rows of $\mathrm{Row}(N)$ form a basis of the row space of N. Replace (4) by (16), do the same perturbation analysis on (9) and (16) as we did from (11) to (15), we conclude that the following quantity needs to be maximized instead:

$$\max_C \ \sigma_{\min}\left(\begin{bmatrix} C \\ \mathrm{Row}(N) \end{bmatrix}\right) > 0. \tag{17}$$

The cost function used in [8] is an approximation to (17). We call it the *crashing-avoidance score*:

$$S_{\text{crash}} = \sigma_{\min}(\begin{bmatrix} C \\ \mathrm{Row}(N) \end{bmatrix}). \tag{18}$$

This number describes how unlikely that the system will crash. It should be greater than zero, such that the nominal robot action will not crash.

B. Disengaging

The second failure mode involves breaking a contact that is supposed to be engaging. Under quasi-static assumption, a contact stays engaged as long as the contact normal force is greater than zero (Section III-C). So we can use the magnitude of the contact normal force to evaluate how much disturbance force can the contact withstand:

$$S_{\text{engage}} = A_z \lambda. \tag{19}$$

The score S_{engage} is a vector with one entry for each contact point. Each element of S_{engage} must be greater than zero. We call S_{engage} the *engaging score*.

C. Avoid Slipping for Sticking Contacts

If the ith contact point is sticking, its contact reaction force must stay within the friction cone. The robustness for maintaining a sticking contact can be measured by the minimal amount of disturbance force required to break the sticking contact into sliding, which is the distance between the contact reaction force and the friction cone edges. Fig. 4 illustrates it in the local contact frame, of which the Z axis points to the outward contact normal. We call this number the *sticking score* for contact i.

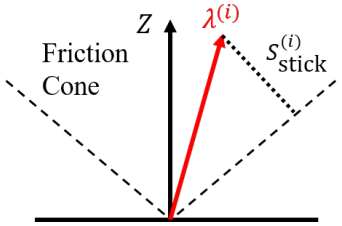

Fig. 4. Definition of $S^{(i)}_{\text{stick}}$ (the dotted line segment) at a contact.

We can compute the score from λ:

$$S^{(i)}_{\text{stick}} = (\mu z^T \lambda^{(i)} - \|z \times \lambda^{(i)}\|) \cos(\arctan \mu). \tag{20}$$

Remember $\lambda^{(i)}$ is the contact force for this particular contact. We assemble the score $S^{(i)}_{\text{stick}}$ for all sticking contacts into a single vector S_{stick}. Again we would like to maximize the scores, and require each of them to be greater than zero. A positive sticking score means the contact reaction force is within the friction cone.

D. Avoid Sticking for Sliding Contacts

Unlike a sticking contact, the robustness for maintaining a sliding contact is tricky to evaluate because the sliding friction is an equality constraint on the contact forces. Unlike the "distance to friction cone" for a sticking contact, there is no concept of "distance towards boundary" for a sliding contact. It's not straightforward to tell whether the contact can stay sliding under modeling uncertainties and force disturbances.

To make the analysis easier, we start from considering one contact point locally, as shown in Fig. 5, right. Describe the location of the ith contact point $P^{(i)} \in \mathbb{R}^3$ in the local contact frame.

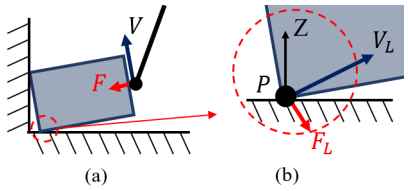

Fig. 5. Left: levering up a block against a corner. Right: close look of one contact point.

The motion of the contact point is determined by two factors. On the one hand, the contact imposes a constraint on the contact point velocity $v_L^{(i)}$:

$$N_L^{(i)} v_L^{(i)} = 0, \tag{21}$$

On the other hand, the motion of the contact point is also subject to a set of force&velocity controls *on this point* $P^{(i)}$, as shown in Fig. 5. These local controls summarize forces&constraints from the robot actions, other contacts on the object and the gravity of the object, etc. The velocity control part imposes a set of linear constraints on $v_L^{(i)}$:

$$c_L^{(i)} v_L^{(i)} = b_{c_L}^{(i)}. \tag{22}$$

1) Ensure sliding for a contact point: Given a set of robot control, we can tell whether the contact is sliding using proof by contradiction:

1) Firstly assume the contact is sticking.
2) Secondly, compute the contact force.
3) If the computed contact force falls in the friction cone, then the contact is indeed sticking. However, if the contact force ends up outside of the friction cone, we find a contradiction and the actual contact must be sliding.

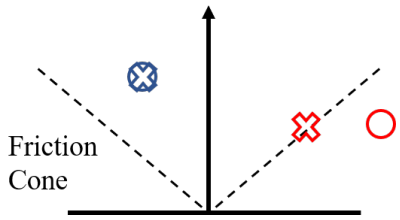

Fig. 6. Contact reaction forces computed by assuming sticking contact (circles), and the corresponding possible actual contact forces (crosses).

This reasoning process above becomes interesting under quasi-static assumption. The result in step three is determined only by the velocity command, since the force along the commanded velocity has infinite magnitude for a sticking contact. If the commanded velocity points outside of the friction cone at a contact point, a finite force command will not be able to move the contact reaction force back into the friction cone; the vice versa. In other words, we ensure sliding robustly regardless of force disturbances by pointing the velocity command outside of the friction cone. Define

$$\psi^{(i)}(c_L^{(i)}, b_{c_L}^{(i)}) = \cos(\arctan \mu) - ||z^T c_L^{(i)\dagger} b_{c_L}^{(i)}||/||c_L^{(i)\dagger} b_{c_L}^{(i)}|| \tag{23}$$

as the angle between the contact force and the friction cone, where $c_L^{(i)\dagger} b_{c_L}^{(i)}$ represents the least-square solution of (22). The condition for sliding is

$$\psi^{(i)}(c_L^{(i)}, b_{c_L}^{(i)}) > 0. \tag{24}$$

Since the friction coefficient is often inaccurate, the safe way to ensure sliding is to point the commanded velocity as close to the contact tangent plane as possible:

$$\max_{c_L^{(i)}, b_{c_L}^{(i)}} \psi^{(i)}(c_L^{(i)}, b_{c_L}^{(i)}). \tag{25}$$

A sliding contact with a positive ψ is also free of jamming [5] because of the velocity command.

2) Conditions for the whole system: Now, instead of looking at a single contact point, let's consider a manipulation system described in Section III-D. In order to compute $\psi^{(i)}$, we need to derive the local constraints $c_L^{(i)}$, $b_{c_L}^{(i)}$ for contact i. In other words, we must know what kind of constraints are imposed on $v_L^{(i)}$ by the velocity command (9) and the natural constraints (4) *except* the constraint from contact i itself. At this point we don't even know whether these influence on $v_L^{(i)}$ are linear constraints.

We approach the problem by examining the solution space of (9) and (4). Denote $\bar{N}^{(i)}$ as N with $N^{(i)}$ removed. Denote $\bar{\Lambda} = \begin{bmatrix} C \\ \bar{N}^{(i)} \end{bmatrix}, b_{\bar{\Lambda}} = \begin{bmatrix} b_C \\ 0 \end{bmatrix}$. Then the solution space to the under-determined linear system $\begin{cases} Cv = b_C \\ \bar{N}v = 0 \end{cases}$ is

$$v = \bar{\Lambda}^\dagger b_{\bar{\Lambda}} + \text{Null}(\bar{\Lambda}) k', \tag{26}$$

where $\bar{\Lambda}^\dagger b_{\bar{\Lambda}}$ is the least-square solution, $\text{Null}(\bar{\Lambda})$ denotes a matrix whose columns form a basis of the null space of $\bar{\Lambda}$, k' is an arbitrary coefficient vector. Project this solution space into the local contact frame, we should have $v_L^{(i)}$:

$$v_L^{(i)} = M^{(i)} v = M^{(i)} \bar{\Lambda}^\dagger b_{\bar{\Lambda}} + M^{(i)} \text{Null}(\bar{\Lambda}) k', \tag{27}$$

This is an affine solution space, the least-square solution is

$$v_L^{(i)*} = (I - (M^{(i)}\text{Null}(\bar{\Lambda}))(M^{(i)}\text{Null}(\bar{\Lambda}))^\dagger) M^{(i)} \bar{\Lambda}^\dagger b_{\bar{\Lambda}}, \tag{28}$$

where all the $\dagger$ indicates pseudo-inverse. Since (27) is an affine solution space, we know the local velocity command is indeed a linear constraint just like (22). The solution space of (22) is supposed to look like the following:

$$v_L^{(i)} = v_L^{(i)*} + \text{Null}(c_L^{(i)}) k', \tag{29}$$

where the first term is the least-square solution given by (28). Comparing the linear part of (29) and (27), we know

$$\text{Null}(c_L^{(i)}) = M^{(i)} \text{Null}(\bar{\Lambda}), \tag{30}$$

so we can compute $c_L^{(i)}$ as a basis of $M^{(i)}\text{Null}(\bar{\Lambda})$'s null space. Note that there are multiple choices for choosing a basis. Any of them is fine because with a suitable choice of $b_{c_L}^{(i)}$ they all correspond to the same velocity command. Here we pick the unit orthogonal basis:

$$c_L^{(i)} = \text{Null}\left((M^{(i)}\text{Null}(\bar{\Lambda}))^T\right) \tag{31}$$

For a sliding contact, the condition for ensuring sliding (24) becomes

$$\psi^{(i)}(C, b_C) > 0, \tag{32}$$

where

$$\psi^{(i)}(C, b_C) = \cos(\arctan \mu^{(i)}) - ||z^T v_L^{(i)*}||/||v_L^{(i)*}||, \tag{33}$$

$v_L^{(i)*}$ is obtained from (28). We denote $S_{\text{slide}}^{(i)} = \psi^{(i)}$ as the *sliding score* for contact i, denote S_{slide} as the concatenation of all $S_{\text{slide}}^{(i)}$. Again a higher score means more robustness.

E. Summary and Usage

A manipulation system with certain desired contact modes may break in four ways: crash due to infeasible velocity constraints; losing contact(s); unexpected slipping or sticking. We can evaluate the robustness of the system against these failures using the crashing-avoidance score S_{crash} (18), the engaging score S_{engage} (19), the sticking score S_{stick} (20) and the sliding score S_{slide} (33). Note that all the scores except S_{crash} have one entry for each corresponding contact point. All the scores must be greater than zero; a negative score indicates a corresponding type of failure on the contact point.

We can use the scores in several ways. For example, we can maximize a weighted sum of the scores to find robot actions in a motion planning or control problem. It is also preferable to constrain all the scores to be greater than zero. The gradient of the crashing-avoidance score S_{crash} (18) is hard to compute, however, we can use a SDP formulation to optimize it [11] or optimize its quadratic approximation as in [8].

V. Examples

We demonstrate how the the robustness scores work on a simple 2D example. Consider a rectangle block lying on the ground. Model the line-to-line contact between the object and the ground by two point contacts at the left and right corner. A robot point finger presses on the top edge of the block to slide it to the right on the ground. The robot action includes one dimensional velocity control and one dimensional force control. The system contains two sliding contacts (on the ground) and one sticking contact (at the point finger). To show the characteristics of the scores clearly, we separate the four scores into force-related scores (determined by the force command) and velocity-related scores (determined by the velocity command). We demonstrate them separately in two test settings, as shown in Fig. 7.

A. Velocity related Scores

In the first setting, we demonstrate the velocity-related scores (crashing-avoidance score S_{crash} (18) and the sliding score S_{slide} (33)). The robot finger presses the center of the object top edge. We test actions whose velocity commands point in different angles, while the corresponding force commands are computed only to satisfy the minimal contact normal force and friction cone constraints. Denote α as the angle between the commanded velocity and the straight-right

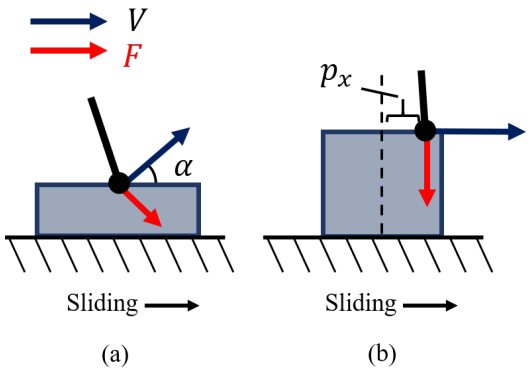

Fig. 7. The two test settings. In test one, we change the angle α of velocity control. In test two, we change the location p_x of the contact point.

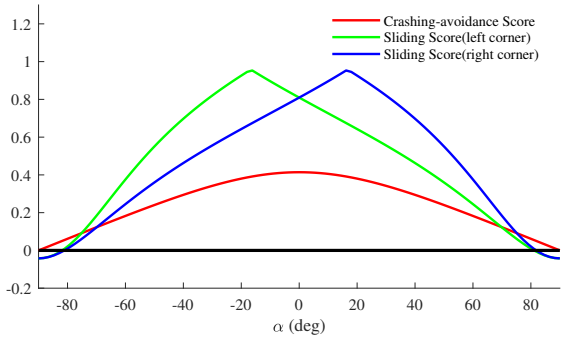

Fig. 8. The velocity-related robustness scores in test setting one.

direction, the velocity-related robustness scores associated with $\alpha = [-90°, 90°]$ are shown in Fig. 8. We omit the force-related scores here because these scores do not change. We can see that when the commanded-velocity points to the contact tangent direction ($\alpha = 0$) the crashing-avoidance score attains the maximum. When the commanded-velocity aligns with the contact normal, penetration happens and the crashing-avoidance score falls to zero accordingly. Note this also includes outwards normal direction, because we model the contact by an equality constraint. This is fine as long as the contact normal force is positive. At the same time the sliding score falls below zero, meaning the left or right corner is going to get stuck.

B. Force-related Scores

In the second setting, we demonstrate the force-related scores, i.e. the engaging score S_{engage} (19) and the sticking score S_{stick} (20). . We fix the direction and magnitude of both force and velocity commands, then compute the scores for different hand contact locations as shown in Fig. 7, right. The velocity-related scores are the same across these settings, so we only show the force-related scores in Fig. 9. The pressing location p_x ranges from the top left corner to the top right corner of the object. Since we are pressing with a constant force, the engaging score and sticking score for the hand contact stay the same. As p_x approaches the right corner, the left corner feels less and less contact normal force and

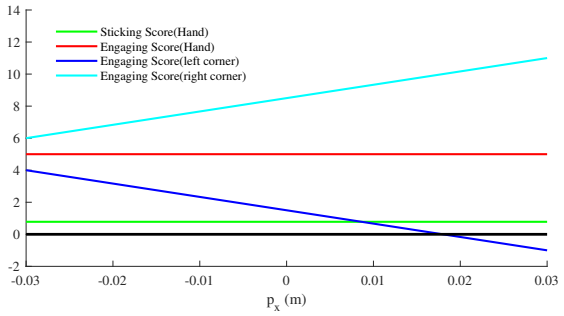

Fig. 9. The force-related robustness scores in test setting two.

will ultimately disengage as its engaging score falls below zero.

An animation showing the change of scores for both settings can be found in the video submission.

VI. Limitations and Future Work

The efficacy of our robustness scores may be affected by deficiencies in the robot hardware. The biggest potential problem comes from the condition for engaging, which require a positive normal force. Consider the example shown in Fig. 7, left. If $\alpha = 80°$, the force control on the robot might not be fast enough to maintain a positive contact force when the velocity control is pulling it away from the contact. For a real robot system, it takes time for a contact force to build up. We can mitigate the problem by requiring a higher crashing-avoidance score, which will encourage the local velocity commands to point closer to the contact tangent plane (reduce α in Fig. 7). Also we can slow down the execution, or implement the force control with faster response.

In the future, the contact robustness criteria can be used in several different aspects of robotic manipulation. At control level, we can improve the original hybrid servoing algorithm in [8] by directly optimizing the robustness scores. We expect to see performance improvements especially in maintaining sliding contacts. At planning level, the robustness criteria can help to find easy-to-execute trajectories. We may optimize the trajectory about the scores, or use the scores to guide the sampling in sampling-based methods. For robot learning, the criteria provide a more informative labeling than success or failure, which could be helpful for faster learning rate.

ACKNOWLEDGMENT

We thank Ankit Bhatia for constructive suggestions.

References

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

[2] Nikhil Chavan-Dafle, Rachel Holladay, and Alberto Rodriguez. In-hand manipulation via motion cones. *Robotics: Science and systems XIV*, 2018.

[3] Nikhil Chavan-Dafle, Matthew T Mason, Harald Staab, Gregory Rossano, and Alberto Rodriguez. A two-phase gripper to reorient and grasp. In *Automation Science and Engineering (CASE), 2015 IEEE International Conference on*, pages 1249–1255. IEEE, 2015.

[4] Mehmet Dogar and Siddhartha Srinivasa. A framework for push-grasping in clutter. *Robotics: Science and systems VII*, 1, 2011.

[5] Pierre E Dupont and Serge P Yamajako. Jamming and wedging in constrained rigid-body dynamics. In *Proceedings of the 1994 IEEE international conference on robotics and automation*, pages 2349–2354. IEEE, 1994.

[6] 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.

[7] Yifan Hou, Zhenzhong Jia, Aaron M Johnson, and Matthew T Mason. Robust planar dynamic pivoting by regulating inertial and grip forces. In *The 12th International Workshop on the Algorithmic Foundations of Robotics (WAFR)*. Springer, 2016.

[8] 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.

[9] Steve Jacobsen, E Iversen, D Knutti, R Johnson, and K Biggers. Design of the utah/mit dextrous hand. In *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, volume 3, pages 1520–1532. IEEE, 1986.

[10] Aaron M Johnson, Jennifer E King, and Siddhartha Srinivasa. Convergent planning. *IEEE Robotics and Automation Letters*, 1(2):1044–1051, 2016.

[11] Shahar Z. Kovalsky, Noam Aigerman, Ronen Basri, and Yaron Lipman. Controlling singular values with semidefinite programming. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)*, 33(4), 2014.

[12] Kevin M Lynch and Matthew T Mason. Stable pushing: Mechanics, controllability, and planning. *The International Journal of Robotics Research*, 15(6):533–556, 1996.

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

[14] Igor Mordatch, Emanuel Todorov, and Zoran Popović. Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics (TOG)*, 31(4):43, 2012.

[15] Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[16] 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.

[17] Michael Posa, Mark Tobenkin, and Russ Tedrake. Lyapunov analysis of rigid body systems with impacts and friction via sums-of-squares. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*, pages 63–72. ACM, 2013.

[18] Michael Posa, Mark Tobenkin, and Russ Tedrake. Stability analysis and control of rigid-body systems with impacts and friction. *IEEE Transactions on Automatic Control (TAC)*, 61(6):1423–1437, 2016.

[19] J Kenneth Salisbury and John J Craig. Articulated hands: Force control and kinematic issues. *The International journal of Robotics research*, 1(1):4–17, 1982.

[20] Jian Shi, J Zachary Woodruff, and Kevin M Lynch. Dynamic in-hand sliding manipulation. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 870–877, 2015.

[21] Jeffrey C Trinkle, Soon-Lin Yeap, and Li Han. When quasistatic jamming is impossible. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 3401–3406. IEEE, 1996.

[22] Francisco E Vi, Yiannis Karayiannidis, Christian Smith, Danica Kragic, et al. Adaptive control for pivoting with visual and tactile feedback. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 399–406.

[23] Miomir Vukobratović and Branislav Borovac. Zero-moment point-thirty five years of its life. *International journal of humanoid robotics*, 1(01):157–173, 2004.

[24] Daniel E Whitney. *Mechanical assemblies: their design, manufacture, and role in product development*, volume 1. Oxford Series on advanced manufacturing, 2004.