In-Hand Manipulation via Motion Cones

Nikhil Chavan-Dafle, Rachel Holladay, and Alberto Rodriguez
Massachusetts Institute of Technology
{nikhilcd, rhollada, albertor}@mit.edu

***Abstract*—In this paper we present the mechanics and algorithms to compute the set of feasible motions of an object pushed in a plane. This set is known as the *motion cone* and was previously described for non-prehensile manipulation tasks in the horizontal plane. We generalize its geometric construction to a broader set of planar tasks, where external forces such as gravity influence the dynamics of pushing, and prehensile tasks, where there are complex interactions between the gripper, object, and pusher. We show that the motion cone is defined by a set of low-curvature surfaces and provide a polyhedral cone approximation to it. We verify its validity with 2000 pushing experiments recorded with motion tracking system.**

Motion cones abstract the algebra involved in simulating frictional pushing by providing bounds on the set of feasible motions and by characterizing which pushes will stick or slip. We demonstrate their use for the dynamic propagation step in a sampling-based planning algorithm for in-hand manipulation. The planner generates trajectories that involve sequences of continuous pushes with 5-1000x speed improvements to equivalent algorithms.

I. Introduction

A *motion cone* is the set of feasible motions that a rigid body can follow under the action of a frictional push. We can think of it as a geometric representation of the underactuation inherent to frictional contacts. Since contacts can only push, and since friction is limited, a contact can move an object only along a limited set of rays. The concept was introduced by Mason [22] for point contacts in the context of a planar horizontal pushing task.

Motion cones abstract the algebra involved in simulating frictional contact dynamics. A contact force on the inside (boundary) of the friction cone produces sticking (slipping) behavior, and leads to motion rays on the inside (boundary) of the motion cone. Lynch and Mason [21] generalized the construction of motion cones to line contacts in a horizontal plane. They used them to plan stable pushing trajectories without the complexities of standard complementarity formulations of contact dynamics [29, 24, 2]. Motion cones have since been the basis of several efficient planning and control strategies for planar manipulations on a horizontal support surface [9, 5, 11, 34, 12].

This paper studies the construction of motion cones in a more general set of planar tasks. In particular, we highlight the case of prehensile manipulation in the vertical plane. In general planar tasks, external forces other than the pusher force (for example, gravity) can alter the dynamics of contact interactions between the pusher, object, and gripper/support-plane. Motion cones efficiently capture the intricate mechanics of these tasks for simulation, planning, and control.

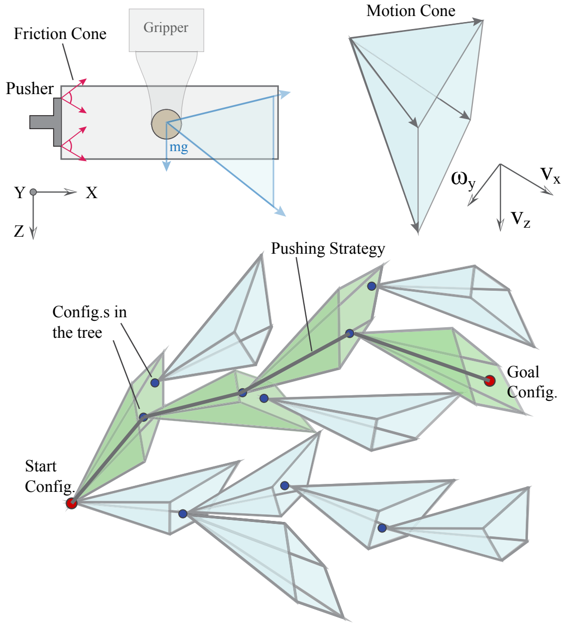

Fig. 1: (top) Example *friction cone* and *motion cone* of an object moving in the vertical plane. The pusher can move the object along any direction $[V_x, V_z, \omega_y]$ inside the motion cone. (bottom) A plan via motion cones. Motion cones capture local reachability. A path in the tree of motion cones generates a pushing strategy to move an object.

We present three main contributions:

- **Mechanics** of motion cones for planar tasks in the gravity plane. We show that the motion cone is defined by a set of low-curvature surfaces, intersecting at a point and pairwise in lines. We propose a polyhedral approximation to the motion cone for efficient computation.
- **Experimental validation** of the stick/slip condition of motion cones in a prehensile pushing task instrumented with a Vicon motion tracker.
- **Application** of motion cones in a sampling-based planning framework for in-hand manipulation using prehensile pushes (see Fig. 1). We show this yields significant speed improvements with respect to our prior work [2, 3].

Fig. 2 shows an example of a pushing trajectory planned to change the grasp on a T-shaped object. The resulting trajectory is a sequence of continuous stable pushes, each where the object sticks to a particular external pusher. The proposed planning algorithm obtains these trajectories consistently in less than a second.

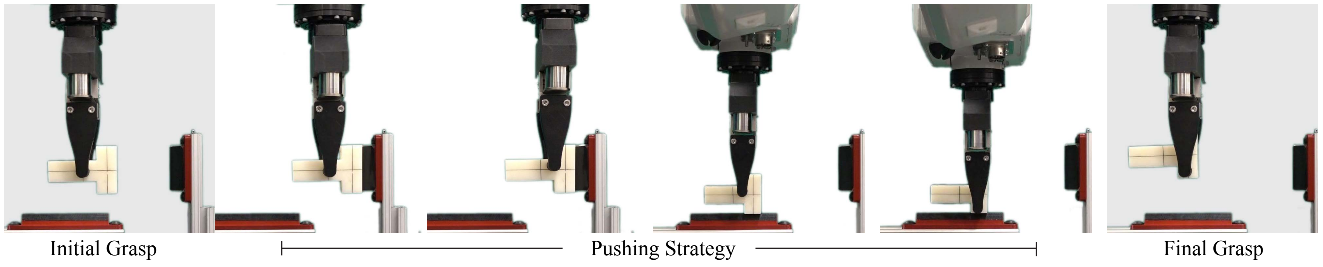

Fig. 2: Manipulating a T-shaped object in a parallel-jaw grasp by pushing it against features in the environment. The manipulation is shown from a side view.

Section III reviews the mechanics of pushing in an arbitrary plane, while Section IV describes the process to construct the corresponding motion cones. In Section V we discuss the computational aspects in the calculation and approximation of motion cones for pushing an object gripped with a finite force. In Section VI we demonstrate a sampling-based planning approach that exploits motion cones to speed up the computation of local contact dynamics for prehensile pushing.

The generalization of motion cones to interactions with gravity opens a door for efficient and robust planning of in-hand manipulations that respect and exploit the basic principles of frictional rigid-body contact interactions: Newton's second law, Coulomb's friction law, the principle of Maximal Dissipation, and the rigidity of rigid-bodies.

II. Related Work

Planning and control through contact is a central topic in robotic manipulation research. Rigid-body contact is modeled as a series of constraints on the possible motions and forces at contact. Simplifying and exploiting these constraints is a pivotal theme in the non-prehensile manipulation literature. Goyal [10] introduced the concept of a *limit surface*, a compact mapping from the friction wrench between slider and ground and the sliding twist at contact. Mason [22] studied the mechanics of pushing and proposed the concept of the *motion cone*. These two fundamental geometric constructions provide direct force-motion mappings for contact interactions and have facilitated efficient planning and control techniques in non-prehensile manipulation [20, 21, 5, 6, 11, 34].

Recent work on trajectory optimization and manipulation planning shows that it is possible to reason about different contact modes and plan trajectories through contact in a standard rigid body dynamics framework based on complementarity constraints. These methods often have to compromise on the computational efficiency [24, 2] or the realism of contact dynamics [19, 31, 18].

In contrast, some more recent work has focused on particular types of manipulation primitives and exploits assumptions in the problem formulation to develop fast planning and robust control strategies. Shi et al. [25] demonstrates dynamic in-hand manipulation planning in a parallel-jaw grasp. With a pre-defined contact mode sequence at the fingers and a limit surface approximation for the force-motion interaction at the fingers, they derive a control law that can move the object to the goal pose in the grasp. Similar approaches are explored for planning and controlling in-hand manipulations by actively using gravity [32] or dynamic motions [13, 28, 14].

Sundaralingam and Hermans [30] propose a trajectory-optimization based, purely-kinematic approach for in-hand manipulation with a multi-finger gripper. They assume that the fingers on the object do not slip and impose soft constraints that in practice minimize the slip at the fingers. With the assumption of sticking at all contacts, and neglecting friction dynamics, they obtain kinematic plans quickly.

In our recent work [3], we present a fast sampling-based planning framework for in-hand manipulations with prehensile pushes, where the pusher contact is forced to stick to the object. The plans are discrete sequences of continuous pushes that respect friction, contact, and rigid-body constraints.

These promising results on in-hand manipulation research, which limit to certain types of manipulation primitives for efficient planing and control, motivate us to generalize the concept of motion cones to more general pushing tasks. Motion cones, as a set of direct constraints on the object motion or control inputs, can naturally fit well in a trajectory optimization framework. As we illustrate in this paper, for sampling-based methods, the motion cone can be used to guide sampling, and for fast dynamics propagation.

III. Mechanics of Pushing in a Plane

Fig. 3 shows four different cases of planar pushing. In case (a), the pusher force is the only external force on the object in the plane of motion. However, in the rest of the cases, a component of gravity is also present. The concept of the motion cone as originally studied in [22] is limited to the case (a) in Fig. 3. Our extension of the motion cone to a general planar case is valid for all the cases in Fig. 3.

In this section we discuss the mechanics of pushing an object in a plane. First we will review the fundamental concepts, namely, the *limit surface* [10], and the *generalized friction cone* [8] that will help us model the friction interaction at the contacts involved in pushing.

A. Limit Surface

The limit surface is a common approach to model a friction interaction between an object and a support contact. In case (a) and (b) in Fig. 3, the surface on which the object rests

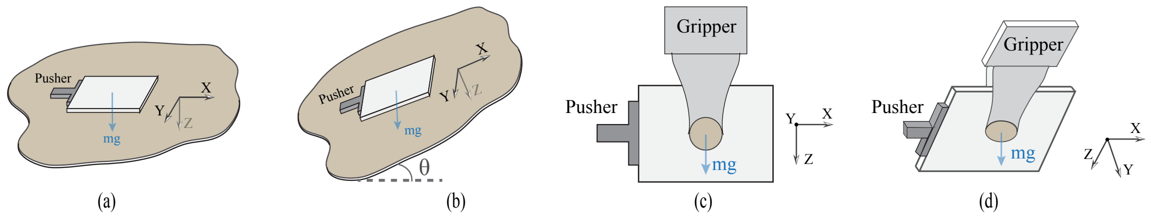

Fig. 3: Pushing an object (a) on a horizontal surface, (b) on an inclined surface, (c) in a grasp in the gravity plane, and (d) in a grasp in a tilted plane

is the support contact. In case (c) and (d), the finger contacts play the same role.

Goyal [10] defined the boundary of the set of all possible friction wrenches that a contact can offer as the *Limit Surface*. Howe and Cutkosky [15], Xydas and Kao [33] showed that an ellipsoidal approximation allows for a simpler representation of the limit surface geometry. In this paper we will assume an ellipsoidal approximation of the limit surface, which has been shown to be computationally efficient for simulating and planning pushing motions [21, 6, 25, 34].

Let $\boldsymbol{w} = [f_{\mathrm{x}}, f_{\mathrm{z}}, m_{\mathrm{y}}]$ be a frictional wrench on the object from the support contact in the contact frame. A mathematical representation of the ellipsoidal limit surface is given by $\boldsymbol{w}^T A \boldsymbol{w} = 1$, where $A = Diag(a_1^{-2}, a_2^{-2}, a_3^{-2})$. For isotropic friction, the maximum friction force is, $a_1 = a_2 = \mu_c N$, where μ_c is the friction coefficient between the contact and the object and N is the normal force at the contact. The maximum friction torque about the contact normal is $a_3 = rc\mu_c N$, where r is the radius of the contact and $c \in [0, 1]$ an integration constant. For a uniform pressure distribution at the contact, c is about 0.6 [33, 25].

When the object slides on the support contact, the friction wrench between the object and the contact ($\boldsymbol{w}_c$) intersects the limit surface. Based on the maximal energy dissipation principle, the normal to the limit surface at the intersection point provides the direction of the twist of the object at the contact. Conversely, if the object twist ($\boldsymbol{v}_{\text{obj}} = [v_x, v_z, \omega_y]^T$) is known, we can find the friction wrench following [25] as,

$$\boldsymbol{w}_c = \frac{A^{-1}\boldsymbol{v}_{\text{obj\_c}}}{\sqrt{\boldsymbol{v}_{\text{obj\_c}}{}^T A^{-1} \boldsymbol{v}_{\text{obj\_c}}}} = \mu_c N \overline{\boldsymbol{w}}_c \qquad (1)$$

Here, $\boldsymbol{v}_{obj\_c}$ is the velocity of the object in the contact frame and can be computed from $\boldsymbol{v}_{\text{obj}}$ as $\boldsymbol{v}_{obj\_c} = \tilde{\boldsymbol{J}}_{\text{c}} \cdot \boldsymbol{v}_{obj}$. The jacobian ($\tilde{\boldsymbol{J}}_{\text{c}}$) maps the object velocity from the object frame to the support contact frame. $\overline{\boldsymbol{w}}_c = [\overline{f}_{\text{x}}, \overline{f}_{\text{z}}, \overline{m}_{\text{y}}]^T$ is the unit wrench lying on a limit surface that is scaled by maximum linear friction available at contact ($\mu_c N$) to produce the net frictional wrench.

Under the ellipsoidal limit surface model assumption, translational velocity $[v_{\text{x\_c}}, v_{\text{z\_c}}]^T$ of the object in the contact frame is always parallel and opposite to the linear frictional force $[\overline{f}_x, \overline{f}_z]^T$ applied by the contact in the contact frame [20]. Moreover, the relationship between the friction wrench and the normal to the limit surface, which defines the motion direction, sets the following constraint between the angular velocity at the contact and the linear velocity:

$$\frac{v_{\text{x\_c}}}{\omega_{\text{y\_c}}} = (rc)^2 \frac{\overline{f}_{\text{x}}}{\overline{m}_{\text{y}}} \quad \text{and} \quad \frac{v_{\text{z\_c}}}{\omega_{\text{y\_c}}} = (rc)^2 \frac{\overline{f}_{\text{z}}}{\overline{m}_{\text{y}}}$$

Given the friction wrench on the object from the support contact, we can find the object velocity as:

$$\boldsymbol{v}_{obj} = b\boldsymbol{J}_{\text{c}}\boldsymbol{B} \cdot \overline{\boldsymbol{w}}_c \ , \ \boldsymbol{B} = Diag(1, 1, (rc)^{-2}), \quad b \in \mathrm{I\!R}^+ \quad (2)$$

Here, $\boldsymbol{J}_{\text{c}}$ maps the object velocity from the support contact frame to the object frame.

B. Generalized Friction Cone

The friction between the pusher and the object can be modelled with the Coulomb friction law. Erdmann [8] introduced the concept of *generalized friction cone* ($\boldsymbol{W}$) as a representation of the local friction cone at a contact in the object frame. The generalized friction cone for a pusher modelled with multiple point contacts is the convex hull of the generalized friction cones for each constituent contact [7].

$$\boldsymbol{W}_{\text{pusher}} = \{\overline{\boldsymbol{w}}_{\text{pusher}} = \boldsymbol{G}_{\text{p}} \cdot \overline{\boldsymbol{f}}_{\text{p}} \mid \overline{\boldsymbol{f}}_{\text{p}} \in FC_{\text{pusher}}\} \qquad (3)$$

Here, $\boldsymbol{G}_{\text{p}}$ is the jacobian that maps the local contact forces ($\boldsymbol{f}_{\text{p}}$) at the pusher to the object frame. $\overline{\boldsymbol{w}}_{\text{pusher}}$ is the unit wrench corresponding to unit force/s $\overline{\boldsymbol{f}}_{\text{p}}$ inside the friction cone/s at the pusher constituent contact/s.

Now, with the approaches for contact modelling set, we look into formulating the mechanics of pushing in a plane.

C. Mechanics of Pushing

The motion of the object in the plane of motion evolves following the net wrench acting on it. Under the quasi-static assumption, which is appropriate for slow pushing operations, the inertial forces on the object are negligible and there is force balance:

$$\boldsymbol{w}_{\text{support}} + \boldsymbol{w}_{\text{pusher}} + m\boldsymbol{g} = 0 \qquad (4)$$

Here, (4) is written in the object frame located at the center of gravity. $\boldsymbol{w}_{\text{support}}$ is the friction wrench provided by the support contact, $\boldsymbol{w}_{\text{pusher}}$ is the wrench exerted by the pusher, m is the mass of the object, $\boldsymbol{g}$ is the gravitational component in the plane of motion. $\boldsymbol{G}_{\text{c}}$ ($\boldsymbol{G}_{\text{c}} = \boldsymbol{J}_{\text{c}}{}^{\top}$) is the jacobian that maps the support contact wrench from the contact frame (usually located at the center of pressure) of the support surface to the object frame. So, $\boldsymbol{w}_{\text{support}} = \boldsymbol{G}_{\text{c}} \cdot \boldsymbol{w}_{\text{c}}$ and (4) becomes:

$$\mu_{\text{c}} N \boldsymbol{G}_{\text{c}} \cdot \overline{\boldsymbol{w}}_{\text{c}} + \boldsymbol{G}_{\text{p}} \cdot \boldsymbol{f}_{\text{p}} + m\boldsymbol{g} = 0 \qquad (5)$$

D. Stable Pushing in a Plane

Lynch and Mason [21] studied pushing motions for which the pusher sticks to the object when pushing on a horizontal surface, which they referred to as *Stable Pushing*. For stable pushing, force at the pusher lies inside the friction cone in the local contact frame.

For a general planar case, such a condition for a stable push can be written as:

$$\mu_{\text{c}} N \boldsymbol{G}_{\text{c}} \cdot \overline{\boldsymbol{w}}_{\text{c}} + \boldsymbol{G}_{\text{p}} \cdot \boldsymbol{f}_{\text{p}} + m\boldsymbol{g} = 0 \ , \ \ \boldsymbol{f}_{\text{p}} \in FC_{\text{pusher}}$$

For a given object motion to be possible with a stable push, the pusher needs to be able to provide a wrench that balances the net wrench produced by the friction wrench from the support contact and the gravitational force in the plane. Using (3) we can rewrite the previous equation as:

$$\begin{gathered} -\mu_{\text{c}} N \boldsymbol{G}_{\text{c}} \cdot \overline{\boldsymbol{w}}_{\text{c}} - m\boldsymbol{g} = k\overline{\boldsymbol{w}}_{\text{pusher}} \\ \overline{\boldsymbol{w}}_{\text{pusher}} \in \boldsymbol{W}_{\text{pusher}} \ , \ \ k \in \mathbb{R}^{+} \end{gathered} \tag{6}$$

Here, k is the magnitude of the pusher force. To know if an object motion can be achieved with a stable push, we simply need to check if the net required wrench falls inside the generalized friction cone of the pusher.

$$-\mu_{\text{c}} N \boldsymbol{G}_{\text{c}} \cdot \overline{\boldsymbol{w}}_{\text{c}} - m\boldsymbol{g} \in \boldsymbol{W}_{\text{pusher}} \tag{7}$$

From a planning and control perspective, rather than querying if pushes are stables pushes or not, the bound on the set of object motions possible with stable pushes is more useful. This is in fact the motivation for the motion cone concept that we study in the next section.

IV. Motion Cone for Planar Pushing

The motion cone is the set of objects motions that can be produced while keeping the pusher contact sticking. In an abstract sense, it is the motion equivalent of the generalized friction cone of the pusher.

Problem: Find the set of object motions for which the net required wrench can be balanced by a wrench inside the generalized friction cone of the pusher.

This is equivalent to finding a set of object motion for which constraint (6) holds true. Rewriting,

$$-\mu_{\text{c}} N \boldsymbol{G}_{\text{c}} \cdot \overline{\boldsymbol{w}}_{\text{c}} - m\boldsymbol{g} = k\overline{\boldsymbol{w}}_{\text{pusher}}, \ \ \overline{\boldsymbol{w}}_{\text{pusher}} \in \boldsymbol{W}_{\text{pusher}}, \ \ k \in \mathbb{R}^{+}$$

$$\begin{gathered} \boldsymbol{G}_{\text{c}} \cdot \overline{\boldsymbol{w}}_{\text{c}} = \frac{k}{-\mu_{\text{c}} N} \overline{\boldsymbol{w}}_{\text{pusher}} + \frac{m}{-\mu_{\text{c}} N} \boldsymbol{g} \\ \overline{\boldsymbol{w}}_{\text{pusher}} \in \boldsymbol{W}_{\text{pusher}} \ , \ \ k \in \mathbb{R}^{+} \end{gathered} \tag{8}$$

Using (2) we can map the support contact wrench $\overline{\boldsymbol{w}}_{\text{c}}$ to the object velocity $\boldsymbol{v}_{obj}$. Hence, to find a motion cone, we first find the set of support contact wrenches ($\overline{\boldsymbol{w}}_{\text{c}}$) that satisfy (8) and then map this wrench-set ($\tilde{\boldsymbol{W}}_{\text{c}}$) to the set of object twists. We denote this object twist-set, i.e. the motion cone, by $\tilde{\boldsymbol{V}}_{\text{obj}}$.

The presence of an external force, (such as the gravitational force) other than the pusher force, in the plane of motion complicates the mechanics and the structure of the motion cone. To explain this effect in detail, we will first consider the case of pushing an object on a horizontal surface where there is no such additional force.

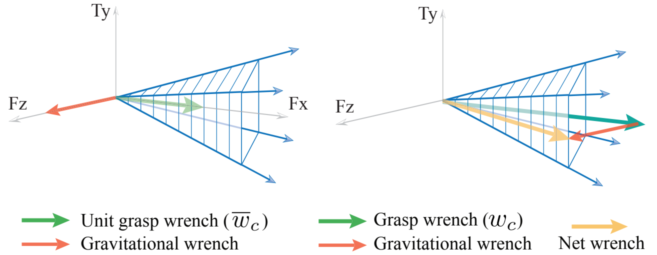

Fig. 4: To make a push inside the gravity-free motion cone stable in the gravity plane, the unit grasp wrench can be scaled such that the net pusher wrench required for the desired push falls inside/on the generalized friction cone of the pusher.

A. Motion Cone for Pushing on a Horizontal Surface

For the case of pushing on a horizontal plane, $\boldsymbol{g} = \boldsymbol{0}$. For an object on a flat support surface with uniform pressure distribution on the support (as is Fig. 3- case (a) and (b)), the contact frame coincides with the object frame, which makes $\boldsymbol{J}_{\text{c}}$, $\tilde{\boldsymbol{J}}_{\text{c}}$, and $\boldsymbol{G}_{\text{c}}$ identity matrices. Then we can write (6) as:

$$\begin{gathered} -\mu_{\text{c}} N \overline{\boldsymbol{w}}_{\text{c}} = k\overline{\boldsymbol{w}}_{\text{pusher}} \ , \ \ \overline{\boldsymbol{w}}_{\text{pusher}} \in \boldsymbol{W}_{\text{pusher}} \ , \ \ k \in \mathbb{R}^{+} \\ \overline{\boldsymbol{w}}_{\text{c}} = \frac{-k\overline{\boldsymbol{w}}_{\text{pusher}}}{\mu_c N} \ , \ \ \overline{\boldsymbol{w}}_{\text{pusher}} \in \boldsymbol{W}_{\text{pusher}} \end{gathered} \tag{9}$$

The set of valid support contact wrenches is the negative of the generalized friction cone of the pusher, i.e, $\tilde{\boldsymbol{W}}_{\text{c}} = -\boldsymbol{W}_{\text{pusher}}$. By mapping $\tilde{\boldsymbol{W}}_{\text{c}}$ through (2), we get the motion cone $\tilde{\boldsymbol{V}}_{\text{obj}}$ [1].

Note that for the case of pushing on a horizontal surface, $\tilde{\boldsymbol{W}}_{\text{c}}$ and $\tilde{\boldsymbol{V}}_{\text{obj}}$ are convex polyhedral cones. Moreover, they are independent of the support normal force, i.e, the weight of the object (mg), and friction at the support surface (μ_c).

For a more general pushing tasks however, $\boldsymbol{g} \neq \boldsymbol{0}$. From (8) we can see that, unlike for horizontal pushing, the system parameter (μ_c) and force magnitudes (k and N) influence the direction vectors of the wrench-set $\tilde{\boldsymbol{W}}_{\text{c}}$. In the next section we will focus on the case shown in Fig. 3-(c) – pushing an object in a parallel-jaw grasp in the plane of gravity. There the gravitational force is not zero in the plane of motion and the jacobians $\boldsymbol{J}_{\text{c}}$, $\tilde{\boldsymbol{J}}_{\text{c}}$, and $\boldsymbol{G}_{\text{c}}$ are not always identity matrices as the support (finger) contact location changes in the object frame as the object is pushed in the grasp. The case shown in Fig. 3-(b) is same as the case (c) except the jacobians $\boldsymbol{J}_{\text{c}}$, $\tilde{\boldsymbol{J}}_{\text{c}}$, and $\boldsymbol{G}_{\text{c}}$ are always identity matrices. The case shown in Fig. 3-(d) is same as the case (c) except that only a part of the gravitational force acts in the plane of motion and not all of the gravitational force as in the case (c).

B. Stable Pushing and Motion Cone in the Gravity Plane

For a case similar to Fig. 3-(c), but in a gravity-free world, we can exploit the simplification of the (8) and compute a convex polyhedral motion cone similar to that in the horizontal pushing case, but while taking the non-identity jacobians

[1] Mason [22] defined the motion cone in terms of pusher twists which is a linear jacobian transform of the motion cone $\boldsymbol{V}_{\text{obj}}$. However, we will keep it in the object twist space.

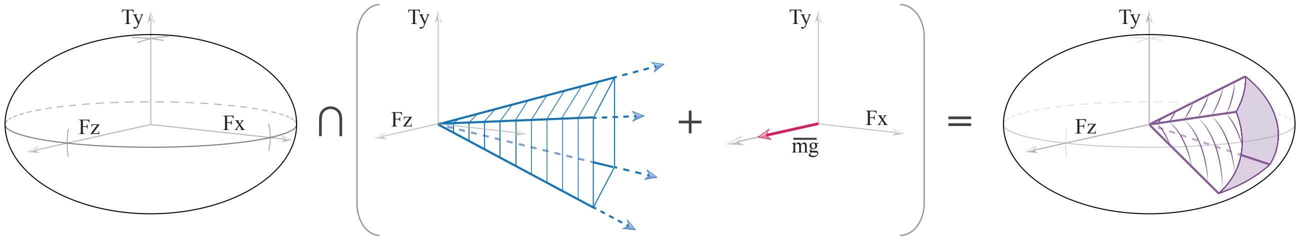

Fig. 5: A depiction of the process for constructing a wrench-set ($\tilde{\boldsymbol{W}}_{\text{c}}$). The intersection of the limit surface with the sum of the scaled generalized friction cone of the pusher and the gravitational wrench defines the wrench-set $\tilde{\boldsymbol{W}}_{\text{c}}$.

$\boldsymbol{J}_{\text{c}}$, $\tilde{\boldsymbol{J}}_{\text{c}}$, and $\boldsymbol{G}_{\text{c}}$ into consideration. We will refer to this motion cone as a gravity-free motion cone in the later discussions.

From (7) we can see that the gravitational wrench ($m\boldsymbol{g}$) on the object can pull the net required wrench in or out of the generalized friction cone of the pusher. Some of the motions in the gravity-free motion cone may not be stable pushes in the gravity plane, while for some object motions outside the gravity-free motion cone, the gravitational force on the object can be exploited to make them stable pushes.

Theorem 1. *Any push inside the gravity-free motion cone can also be made a stable push in the gravity plane by increasing the grasping force above a minimum force threshold.*

Proof. For a motion inside a gravity-free motion cone, the support/grasp wrench direction lies inside the generalized friction cone of the pusher, i.e., $\boldsymbol{G}_{\text{c}} \cdot \overline{\boldsymbol{w}}_{\text{c}} \in \boldsymbol{W}_{\text{pusher}}$. We can always find a magnitude ($\mu_c N$) with which the support/grasp wrench direction needs to be scaled so that the net wrench (the vector sum of the gravitation wrench and the support/grasp wrench) is just inside the generalized friction cone of the pusher. For a given μ_c, we can analytically find the bounds on N needed to pull the net wrench inside the generalized friction cone of the pusher. Fig. 4 shows the graphical interpretation of this proof. □

Theoretically, we can make any motion in the gravity-free motion cone a stable motion in the gravity case by changing the grasping force. With increased grasping force, more motions in a gravity-free motion cone are stable in the gravity plane. For infinite grasping force, the gravity-free motion cone coincides with the motion cone in the gravity plane. However, practically it is not possible for a gripper to provide arbitrarily high grasping force. We need to find the motion cone for a fixed grasping force.

V. Motion Cone in The Gravity Plane for Fixed Grasping Force

In this section we find the object motion cone in the gravity case for a given grasping force and friction parameters.

A. Analytical Computation

Rewriting (8),

$$\boldsymbol{G}_{\text{c}} \cdot \overline{\boldsymbol{w}}_{\text{c}} = \frac{k}{-\mu_{\text{c}} N} \overline{\boldsymbol{w}}_{\text{pusher}} + \frac{m}{-\mu_{\text{c}} N} \boldsymbol{g} \qquad (10)$$
$$\overline{\boldsymbol{w}}_{\text{pusher}} \in \boldsymbol{W}_{\text{pusher}} \;,\;\; k \in \mathbb{R}^{+}$$

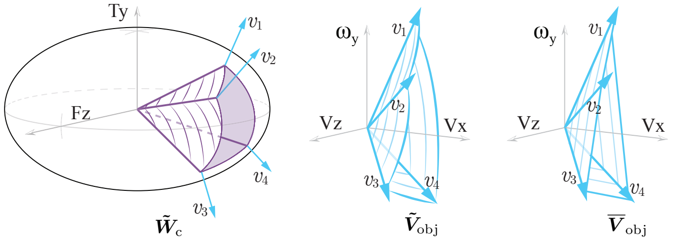
Ty
v_1
v_2
Fz
v_4
v_3
$\tilde{\boldsymbol{W}}_{\text{c}}$
ωy
v_1
v_2
Vz
Vx
v_3
v_4
$\tilde{\boldsymbol{V}}_{\text{obj}}$
ωy
v_1
v_2
Vz
Vx
v_3
v_4
$\overline{\boldsymbol{V}}_{\text{obj}}$

Fig. 6: A graphical illustration for the construction of the motion cone ($\tilde{\boldsymbol{V}}_{\text{obj}}$) from the wrench-set ($\tilde{\boldsymbol{W}}_{\text{c}}$). The motion cone is defined by the set of the surface normals to the limit surface where the wrench-set intersects the limit surface. We propose polyhedral approximation ($\overline{\boldsymbol{V}}_{\text{obj}}$) to the motion cone for computational efficiency.

For a known $\overline{\boldsymbol{w}}_{\text{pusher}} \in \boldsymbol{W}_{\text{pusher}}$, (10) is a set of three linear equalities with 4 unknowns, $\overline{\boldsymbol{w}}_{\text{c}} \in \mathbb{R}^3$ and k. However, we know that $\overline{\boldsymbol{w}}_{\text{c}} = [\overline{f}_{\text{x}}, \overline{f}_{\text{z}}, \overline{m}_{\text{y}}]^T$ is a unit wrench that satisfies the ellipsoidal limit surface constraint:

$$\frac{\overline{f}_{\text{x}}}{1} + \frac{\overline{f}_{\text{z}}}{1} + \frac{\overline{m}_{\text{y}}}{(rc)^2} = 1 \qquad (11)$$

Constraints (10) and (11) can be solved together analytically to find $\overline{\boldsymbol{w}}_{\text{c}}$ and k. Specifically, after substituting $\overline{f}_{\text{x}}, \overline{f}_{\text{z}}$, and $\overline{m}_{\text{y}}$ from (10) into (11), (11) becomes a quadratic equation in k. Solving this quadratic equation for $k \in \mathbb{R}^{+}$ gives a unique solution for k. Substituting this value for k in (10) makes (10) a set of three linear equalities with three unknowns $[\overline{f}_{\text{x}}, \overline{f}_{\text{z}}, \overline{m}_{\text{y}}]$ which can be solved for a unique solution.

For prehensile pushing in the gravity plane, the relationship between $\tilde{\boldsymbol{W}}_{\text{c}}$ and $\boldsymbol{W}_{\text{pusher}}$ is not linear as in the gravity-free case. To find wrench-set $\tilde{\boldsymbol{W}}_{\text{c}}$, we need to sweep $\overline{\boldsymbol{w}}_{\text{pusher}}$ over the boundary of $\boldsymbol{W}_{\text{pusher}}$ and solve (10) and (11) iteratively. Fig. 5 is the depiction of the process involved in solving the constraints (10) and (11) together for computing the wrench cone $\tilde{\boldsymbol{W}}_{\text{c}}$. Fig. 6 shows the graphical representation of the computation in (2) that maps the wrench-set $\tilde{\boldsymbol{W}}_{\text{c}}$ to the motion cone $\tilde{\boldsymbol{V}}_{\text{obj}}$.

$\tilde{\boldsymbol{W}}_{\text{c}}$ and $\tilde{\boldsymbol{V}}_{\text{obj}}$ are not polyhedral cones as in the gravity-free case, but rather can be best characterized as cones defined by low-curvature surfaces that intersect all in a point and pairwise in lines. Fig. 7 shows the wrench cone and motion cone computed analytically for one particular grasp-pusher configuration.

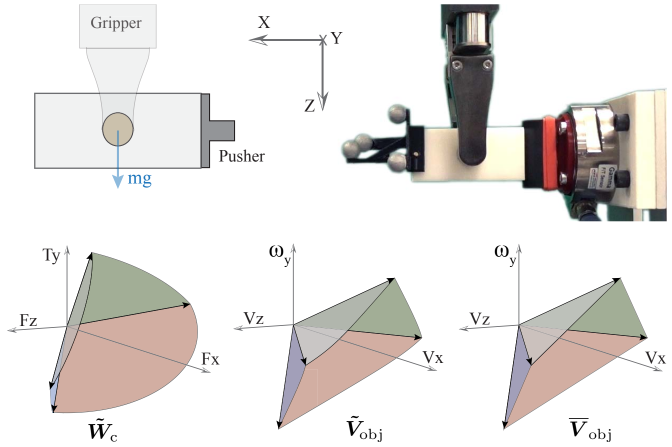
Gripper
X
Y
Z
Pusher
mg
Ty
Fz
Fx
$\tilde{\boldsymbol{W}}_{\text{c}}$
ωy
Vz
Vx
$\tilde{\boldsymbol{V}}_{\text{obj}}$
ωy
Vz
Vx
$\overline{\boldsymbol{V}}_{\text{obj}}$

Fig. 7: (top) The grasp-pusher configuration used for the experimental validation of the motion cone. (bottom) The analytically computed wrench cone ($\tilde{\boldsymbol{W}}_{\text{c}}$), motion cone ($\tilde{\boldsymbol{V}}_{\text{obj}}$), and polyhedral approximation to the motion cone ($\overline{\boldsymbol{V}}_{\text{obj}}$) for the above configuration and 45 N grasping force. Note that the surfaces defining the motion cone ($\tilde{\boldsymbol{V}}_{\text{obj}}$) are curved.

B. Polyhedral Approximation to the Motion Cone

As an object is pushed in a grasp, the position of finger contacts in the object frame change, and consequently $\boldsymbol{G}_{\text{c}}$ and the motion cone $\tilde{\boldsymbol{V}}_{\text{obj}}$ also change. We need to compute the motion cone iteratively as the object moves in the grasp.

As the boundary surfaces of the motion cone have low curvatures, we propose a polyhedral approximation ($\overline{\boldsymbol{V}}_{\text{obj}}$) of the motion cone for its efficient computation. Each edge of the polyhedral approximation of the motion cone is the object motion corresponding to each edge of the generalized friction cone of the pusher. Fig. 6 and Fig. 7 show the polyhedral approximation of the motion cone.

Procedure to compute polyhedral motion cone:

1. Solve (10) and (11) simultaneously to get $\overline{\boldsymbol{w}}_{\text{c}}$ for $\overline{\boldsymbol{w}}_{\text{pusher}}$ corresponding to every edge of $\boldsymbol{W}_{\text{pusher}}$.
2. Define the set of $\overline{\boldsymbol{w}}_{\text{c}}$ computed in step 1 as the generators/edges of the support/grasp wrench-cone $\overline{\boldsymbol{W}}_{\text{c}}$.
3. Map $\overline{\boldsymbol{W}}_{\text{c}}$ to the object twist space using (2) to get the polyhedral approximation $\overline{\boldsymbol{V}}_{\text{obj}}$ of the motion cone.

C. Experimental Validation of the Motion Cone

We use a manipulation platform equipped with an industrial robot arm, a parallel-jaw gripper, a feature in the environment that act as pusher, and a Vicon system for object tracking.

To evaluate the experimental validity of the motion cone in the gravity plane, we collected data for the slip observed at the pusher contact for 2000 randomly sampled pushes. We used the rectangular prism object listed in Table I and the grasping force was set to 45 N. Fig. 7 shows the grasp-pusher configuration used for the experiments. Due to kinematic constraints of the robot and the workspace, we limit the sampled pushes to a small space of $[0\text{ mm-}10\text{ mm}, -5\text{ mm-}5\text{ mm}, -35°\text{- }35°]$ in $[X, Z, \theta_Y]$.

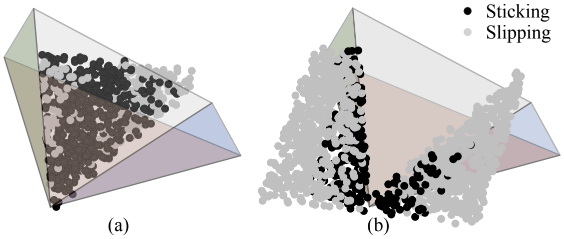

Fig. 8: We carried out 2000 random prehensile pushes in the configuration shown in Fig. 7 and characterized them by the slip observed at the pusher contact. The motion cone above is the same polyhedral motion cone ($\overline{\boldsymbol{V}}_{\text{obj}}$) computed in Fig. 7, but shown in a different orientation for better visualization. (a) No slipping was observed for most of the pushes inside the polyhedral approximation of the motion cone, (b) Most of the pushes for which the slipping was observed are outside the motion cone.

Fig. 8 shows that the polyhedral approximation of the motion cone captures most of the pushes for which the pusher contact sticks to the object. Most of the pushes that slip, fall outside the cone.

VI. Planning In-Hand Manipulations via Motion Cones

In this section we demonstrate the application of motion cones for planning in-hand manipulations with stable prehensile pushes. The problem formulation is similar to the one presented in our recent work [3]. We assume that an object is grasped in a parallel-jaw gripper and manipulated in the gravity plane by pushing against features in the environment. We assume the following physical properties of the system:

- Object geometry and mass.
- Initial and goal pose of an object in a grasp, specified by the locations of each finger contacts.
- Gripping force.
- Set of pusher contacts, specified by their geometries and locations in the object frame.
- Coefficient of friction at all contacts.

In [3], we present a planning framework where at the high-level, a T-RRT*-based architecture samples different object poses in the grasp. At the low level, a rejection check is implemented using a constraint similar to (7) to evaluate if the sampled configuration can be reached using a stable prehensile push. We adopt the same high-level T-RRT*-based planning approach in this paper, but without the low-level rejection check. Rather, the planner always grows the tree towards the sampled pose as best as possible using the motion cones at the nearest node to the sampled pose.

The high-level planning framework is based on T-RRT* - an optimal sampling based method developed for planning on configuration space cost-maps [4, 16].

For selective exploration, the TRRT* framework relies on a transition test that filters the sampled configurations to prefer exploration in low configuration-cost regions. We define the configuration cost as the distance from the goal. The transition test softly constrains the stochastic exploration towards the goal grasp, while allowing the flexibility to explore high-cost transitions if they are necessary to get the object to the goal.

For effective connections, the T-RRT* algorithm uses the underlying RRT* [17] framework to make and rewire the connections in the tree at every step such that the cost of the nodes is reduced when possible. We define the cost of a node as the sum of the cost of the parent node and the cost of the push to reach the sampled node from the parent node. The cost of a push is 0.1 if the parent node uses the same pusher as the child and 1 otherwise. With our node cost definition, the planner generates pushing strategies that prefer fewer pusher switch-overs to push the object to the desired pose.

Algorithm 1 : In-Hand Manipulation Planner

```
input : $q_{init}, q_{goal}$
output : tree $\mathcal{T}$
  $\mathcal{T} \leftarrow$ initialize tree($q_{init}$)
  generate_motionCones($\mathcal{T}, q_{init}$)
  while $q_{goal} \notin \mathcal{T}$ or cost($q_{goal}$) $>$ cost threshold do
      $q_{rand} \leftarrow$ sample random configuration($\mathcal{C}$)
      $q_{parent} \leftarrow$ find nearest neighbor($\mathcal{T}, q_{rand}$)
      $q_{sample} \leftarrow$ take unit step($q_{parent}, q_{rand}$)
      if $q_{sample} \notin \mathcal{T}$ then
          if transition test($q_{parent}, q_{sample}, \mathcal{T}$) then
              $q_{new} \leftarrow$ motionCone_push($q_{parent}, q_{sample}$)
              if transition test($q_{parent}, q_{new}, \mathcal{T}$) and
                grasp maintained($q_{new}$) then
                  $q*_{parent} \leftarrow$ optimEdge($\mathcal{T}, q_{new}, q_{parent}$)
                  add new node($\mathcal{T}, q_{new}$)
                  add new edge($q*_{parent}, q_{new}$)
                  generate_motionCones($\mathcal{T}, q_{new}$)
                  rewire tree($\mathcal{T}, q_{new}, q*_{parent}$)
```

Let q denote a configuration of an object, i.e., the pose of the object in the gripper frame, which is fixed in the world. In this paper, we are considering planar manipulations in a parallel-jaw grasp, so the configuration space $\mathcal{C}$ is $[X, Z, \theta_y] \in \mathbb{R}^3$, i.e., the object can translate in the grasp plane (XZ) and rotate about a perpendicular (Y) to the grasp plane.

Algorithm 1 shows our in-hand manipulation planner. Let q_{init} and q_{goal} be an initial and desired pose of the object in the gripper frame respectively. The planner initiates a tree $\mathcal{T}$ with q_{init} and generates motion cones at q_{init}.

While the desired object pose is not reached within some cost threshold, a random configuration (q_{rand}) is sampled. A nearest configuration (q_{parent}) to q_{rand} in the tree $\mathcal{T}$ is found and an unit-step object pose (q_{sample}) towards q_{rand} is computed. Using the transition test, the planner evaluates if moving in the direction of q_{sample} from q_{parent} is beneficial or not. If it is beneficial, the *motionCone_push* routine computes an object configuration (q_{new}) closest to q_{sample} that can be reached using the motion cones at q_{parent}. It is further checked if moving towards q_{new} is beneficial and if q_{new} is an object configuration at which the grasp on the object is maintained. If both the criteria are satisfied, q_{new} is added to the tree such that the local cost of q_{new} and the nodes near q_{new} are lowered if possible. The motion cones are generated for every new node added to the tree.

Two important routines in Algorithm 1, particularly for this paper, are *generate_motionCones* and *motionCone_push*.

generate_motionCones computes polyhedral motion cones for a given object configuration in the grasp using the procedure listed in Section V-B. At every node, we will have the same number of motion cones as that of pushers.

motionCone_push finds an object pose closest to the desired sampled pose (q_{sample}) that can be reached. This computation is done using the motion cones at the parent node (q_{parent}). If the object twist needed from the parent node pose to the sampled pose is already inside any of the motion cones, the sampled pose can be directly reached. If the required object twist is outside all the motion cones, a twist that is inside one of the motion cones and closest to the desired twist is selected.

The use of motion cones for fast low-level unit-step propagation of the system and T-RRT*-based framework for high-level planning allows us to explore the configuration space of different object poses in the grasp and generate pushing strategies for the desired in-hand manipulation.

VII. Regrasp Examples and Experimental Results

We evaluate the performance of our planner with examples of a parallel-jaw gripper manipulating a variety of objects. The initial pose of an object in the gripper is treated as $[X, Z, \theta_Y] = [0, 0, 0]$. Table II lists the goal poses (in [mm, mm, deg.]) for different examples. While there are no comparable available algorithms that can solve the type of regrasps we are interested in, we provide comparisons with our own implementations of the same high-level planner paired with different algorithms to solve the mechanics of prehensile pushing. These include sampling with rejection by a feasibility check for stable pushing [3], and a complementarity formulation (MNCP) that allows both sticking and slipping at the pusher contact [2]. We compare the performance in terms of planning time and the quality of the solutions. The planning times in Table II are the median times over 10 trials. All the computations are done in MATLAB R2017a on a computer with Intel Core i7 2.8 GHz processor.

TABLE I: Physical properties of the experimental objects

Shape	**Material**	**Dim [L, B, H]** (mm)	**Mass** (g)
square prism	Al 6061	100, 25, 25	202
rectangular prism	Delrin	80, 25, 38	113
T-shaped	ABS	70, 25, 50	62

TABLE II: Planning times (sec.) for approaches using motion cone, stable check [3] and MNCP [2] for unit-step propagation

Manipulation	**Goal**	**Planning**	**Planning**	**Planning**
	$[X, Z, \theta_Y]$	**Time**	**Time** [3]	**Time** [2]
Horz. offset (low μ)	20, 0, 0	0.45	2.83	592.8
$[X, Z, \theta_Y]$ Regrasp	15, -13, 45	0.67	2.54	17684
T-shaped	25, 17.5, 0	0.54	0.82	32657

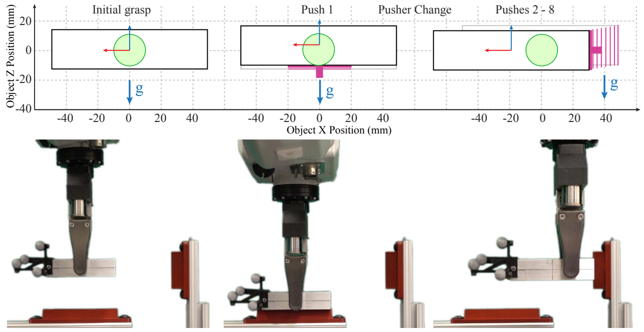

Fig. 9: Simulation and experimental run for a pushing strategy to regrasp the aluminum object with low friction pushers. In the simulation figure (top), the finger and pusher contacts are shown in green and magenta color respectively.

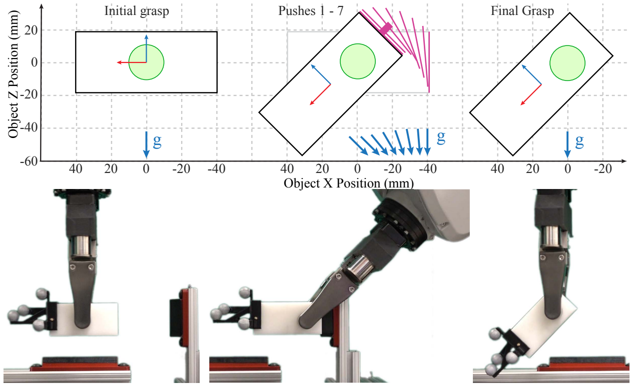

Fig. 10: A pushing strategy for $[X, Z, \theta_Y]$ regrasp. In simulation, the direction of gravity remains constant in the pusher frame, because in reality, the pushers are fixed features.

In all the examples below, we assume three line pushers on the object, one on each side faces of the object parallel to the Z axis and one under the object parallel to the X axis. We use high friction line pushers, except in the first example.

1) Regrasping an object offset to the center: In this example, the goal is to regrasp the square prism horizontally 20 mm offset from the center. We use low friction pushers first. Kolbert et al. [18] showed that for a similar setting, if the object is pushed horizontally in the grasp it slides down as it moves sideways in the grasp. For the low-friction pushers, our planner generates a strategy where the object is first pushed up using the bottom pusher and then the side pusher is used to virtually keep the object stationary while the fingers slide up and along the length of the object as seen in Fig. 9. This plan is similar to the one found in [2, 3].

When we replace the pushers with high-friction pushers (pushers with rubber coating), the planner detects that the desired object twist lies inside the motion cone for the side pusher at the initial grasp pose, i.e, simply pushing from the side is a valid pushing strategy.

2) Regrasp in $[X, Z, \theta_Y]$: The goal in this example is to regrasp the rectangular prism requiring twist in all three dimensions $[X, Z, \theta_Y]$. Similar to [3], our planner finds a strategy to achieve the regrasp using only one pusher. In fact, as we can see in Fig. 10, the pushing strategy our planner comes up with is more direct and seems to avoid unnecessary object motions seen in the strategy shown in [3]-Fig.1.

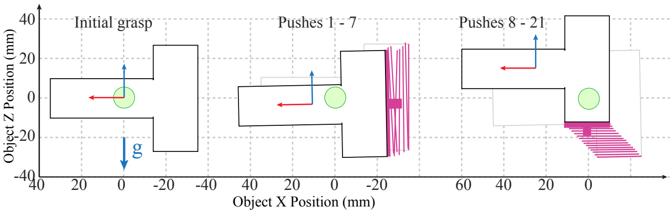

Fig. 11: Simulated motion of the object in the grasp for a pushing strategy to manipulate T-shaped object. Snapshots of the experimental run are is shown in Fig. 2

3) Manipulating a non-convex object: In this example, the goal is to regrasp a T-shaped object. The goal pose is such that a greedy approach to push the object directly towards the goal will result in losing the grasp on the object. Our planner comes up with a pushing strategy that respects the geometric constraints of the problem as shown in Fig. 11.

VIII. Discussion

The motion cone is the set of possible motions (twists) of an object pushed in a plane. It also describes the set of motions of the pusher that yield sticking behavior. It abstracts away the complex dynamics of frictional pushing and provides direct bounds in the action (or effect) space for pushing tasks.

In this paper we extend the concept of motion cones to a general set of planar pushing tasks with external forces such as the gravitational force in the plane of motion. We show that the motion cone for a general planar push is defined as a cone with low-curvature faces, and propose a polyhedral approximation for efficient computation.

We demonstrate the use of motion cones as the propagation step in a sampling-based planner for in-hand manipulation. Combining a T-RRT*-based high level planning framework and a *motion cone*-based dynamics propagation, the planner builds in-hand manipulation strategies with sequences of continuous prehensile pushes in a fraction of a second.

The motion cone provides a direct knowledge of the set of reachable configurations. Such a structure of reachable volumes could enable planning through regions/volumes of configuration space [1, 23, 26, 27]. Moreover, motion cones, as bounds on pusher actions or as bounds on the object motions, have an adequate form to be incorporated into trajectory optimization frameworks to plan pushing strategies. We believe that the extension and application of motion cones to more general settings provides new opportunities for fast and robust manipulation through contact.

Acknowledgments

This material is based upon the work supported by Mathworks, the Chang Award and the MIT Merrill Lynch Fellowship. We would like to thank the members of the Mcube Lab for helpful discussion and advice.

References

[1] Oliver Brock and Lydia E Kavraki. Decomposition-based motion planning: A framework for real-time motion planning in high-dimensional configuration spaces. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1469–1474, 2001.

[2] Nikhil Chavan-Dafle and Alberto Rodriguez. Sampling-based Planning of In-Hand Manipulation with External Pushes. In *Int Symp Robot Res*, 2017.

[3] Nikhil Chavan-Dafle and Alberto Rodriguez. Stable Prehensile Pushing: In-Hand Manipulation with Alternating Sticking Contacts. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[4] D. Devaurs, T. Siméon, and J. Cortés. Optimal Path Planning in Complex Cost Spaces With Sampling-Based Algorithms. *IEEE Transactions on Automation Science and Engineering*, 13(2):415–424, 2016.

[5] M. R. Dogar and S. S. Srinivasa. Push-grasping with dexterous hands: Mechanics and a method. In *IEEE International Conference on Intelligent Robots and Systems*, pages 2123–2130, 2010.

[6] Mehmet Dogar and Siddhartha Srinivasa. A Framework for Push-grasping in Clutter. In *Robotics: Science and Systems*. MIT Press, July 2011.

[7] M. Erdmann. Multiple-point contact with friction: Computing forces and motions in configuration space. In *IEEE Int Conf Robot and Sys*, volume 1, pages 163–170 vol.1, 1993.

[8] Michael Erdmann. On a Representation of Friction in Configuration Space. *International Journal of Robotics Research*, 13(3):240–271, 1994.

[9] Michael Erdmann. An Exploration of Nonprehensile Two-Palm Manipulation. In *International Journal of Robotics Research*, volume 17, 1998.

[10] S Goyal. *Planar sliding of a rigid body with dry friction: Limit surfaces and dynamics of motion*. Phd dissertation, Department of Mechanical Engineering, Cornell University, 1989.

[11] Francois Hogan and Alberto Rodriguez. Feedback control of the pusher-slider system: a story of hybrid and underactuated contact dynamics. In *12th Int Workshop on the Algorithmic Foundations of Robotics*, 2016.

[12] Francois Hogan, Eudald Romo Grau, and Alberto Rodriguez. Reactive Planar Manipulation with Convex Hybrid MPC. In *IEEE International Conference on Robotics and Automation*, 2018.

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

[14] Yifan Hou. Robust Planar Dynamic Pivoting by Regulating Inertial and Grip Forces. Master's thesis, Robotics Institute , Carnegie Mellon University, Pittsburgh, PA, May 2017.

[15] Robert D. Howe and Mark R. Cutkosky. Practical Force-Motion Models for Sliding Manipulation. *The International Journal of Robotics Research*, 15(6):557–572, 1996.

[16] L. Jaillet, J. Cortés, and T. Siméon. Sampling-Based Path Planning on Configuration-Space Costmaps. *IEEE Transactions on Robotics*, 26(4):635–646, 2010.

[17] Sertac Karaman and Emilio Frazzoli. Sampling-based Algorithms for Optimal Motion Planning. *International Journal of Robotics Research*, 30(7):846–894, 2011.

[18] Roman Kolbert, Nikhil Chavan-Dafle, and Alberto Rodriguez. Experimental Validation of Contact Dynamics for In-Hand Manipulation. In *International Symposium on Experimental Robotics*, 2016.

[19] V. Kumar, Y. Tassa, T. Erez, and E. Todorov. Real-time behaviour synthesis for dynamic hand-manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6808–6815, 2014.

[20] K. M. Lynch, H. Maekawa, and K. Tanie. Manipulation And Active Sensing By Pushing Using Tactile Feedback. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 416–421, 1992.

[21] Kevin M. Lynch and Matthew T. Mason. Stable Pushing: Mechanics, Controllability, and Planning. *The International Journal of Robotics Research*, 15(6):533–556, 1996.

[22] Matthew T. Mason. Mechanics and Planning of Manipulator Pushing Operations. *The International Journal of Robotics Research*, 5(3):53–71, 1986.

[23] Marco Morales, Roger Pearce, and Nancy M Amato. Analysis of the evolution of C-Space models built through incremental exploration. In *IEEE International Conference on Robotics and Automation*, pages 1029–1034, 2007.

[24] Michael Posa, Cecilia Cantu, and Russ Tedrake. A Direct Method for Trajectory Optimization of Rigid Bodies Through Contact. *Int Journal of Robotics Res.*, 33(1): 69–81, 2014.

[25] J. Shi, J. Z. Woodruff, and K. M. Lynch. Dynamic in-hand sliding manipulation. In *2015 IEEE International Conference on Intelligent Robots and Systems*, pages 870–877, 2015.

[26] A. Shkolnik, M. Walter, and R. Tedrake. Reachability-guided sampling for planning under differential constraints. In *IEEE International Conference on Robotics and Automation*, pages 2859–2865, 2009.

[27] Alexander Shkolnik and Russ Tedrake. Sample-Based Planning with Volumes in Configuration Space. In *Available on arXiv*, 2011.

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

[29] D. E. Stewart and J. C. Trinkle. An Implicit Time-Stepping Scheme for Rigid Body Dynamics with Inelastic Collisions and Coulomb Friction. *Intl Journal for*

Numerical Methods in Engineering, 39(15):2673–2691, 1996.

[30] Balakumar Sundaralingam and Tucker Hermans. Relaxed-Rigidity Constraints: In-Grasp Manipulation using Purely Kinematic Trajectory Optimization. In *Proceedings of Robotics: Science and Systems*, 2017.

[31] E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

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

[33] Nicholas Xydas and Imin Kao. Modeling of Contact Mechanics and Friction Limit Surfaces for Soft Fingers in Robotics, with Experimental Results. *The International Journal of Robotics Research*, 18(9):941–950, 1999.

[34] Jiaji Zhou and Matthew T. Mason. Pushing revisited: Differential flatness, trajectory planning and stabilization. In *Int Symp Robot Res*, 2017.