

A Global Quasi-Dynamic Model for Contact-Trajectory Optimization

Bernardo Aceituno-Cabezas¹ and Alberto Rodriguez¹

¹Department of Mechanical Engineering — Massachusetts Institute of Technology

<aceituno,albertor>@mit.edu

Abstract—Given a desired object trajectory, how should a robot make contact to achieve it? This paper proposes a global optimization model for this problem with alternated-sticking contact, referred to as **Contact-Trajectory Optimization**. We achieve this by reasoning on simplified geometric environments with a quasi-dynamic relaxation of the physics. These relaxations are the result of approximating bilinear torque effects and deprecating high-order forces and impacts. Moreover, we apply convex approximations that retain the fundamental properties of rigid multi-contact interaction. As result, we derive a mixed-integer convex model that provides global optimality, infeasibility detection and convergence guarantees. This approach does not require seeding and accounts for the shapes of the object and environment. We validate this approach with extensive simulated and real-robot experiments, demonstrating its ability to quickly and reliably optimize multi-contact manipulation behaviors.

I. INTRODUCTION

Since the early years of robotics, resolving contact interaction has hindered the deployment of robots for general manipulation tasks. Contact has been, and still is, difficult to observe, predict and control. The last decade has seen significant advance in planning-through-contact tools for general motion generation. Their wide-spread use in robotics, however, has been limited by coarse approximations of the mechanics of contact, or by numerical difficulties that arise from the choice of problem formulation – e.g., hybrid dynamics, non-unique solutions, non-convex constraints. These consequences are difficult to avoid in a problem often specified as the optimization of the interaction between a robot and an object, both free variables, to achieve a loosely specified behavior.

In this work we focus on a different perspective: We assume as input a desired trajectory of the object in its environment, and ask how, where, and when should the robot make contact to achieve it. We refer to this problem as **Contact-Trajectory Optimization**, in contrast to the more general trajectory optimization problem, where the object trajectory is also a free variable [30]. This sub-problem retains some of the key challenging aspects of planning through contact, namely hybridness and non-convexity. We will see, however, that this provides multiple benefits. It leads to a better defined problem, with more numerically stable solutions, and ultimately a global model for planning-through-contact with certificates: global optimality, global infeasibility and convergence.

In order to achieve this goal, we rely on a quasi-dynamic approximation of the motion equations and polygonal (convex or non-convex) descriptions of objects and environments.

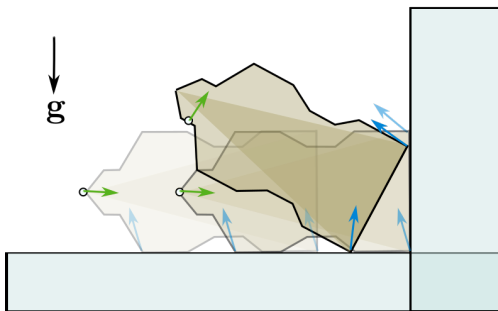


Fig. 1: By reasoning about an approximate world, with quasi-dynamics and polytopic shapes, we reliably optimize alternated-sticking contact interactions to achieve complex manipulation tasks. Our model receives an object motion and outputs a contact-trajectory (positions and forces).

The term quasi-dynamics refers to an approximation of the equations that describe the time-evolution of the system that ignores high-order inertial terms, assumes uniform pressure contact distributions, and exploits a mixed integer-convex approximation of nonlinear relations. In contrast to previous work that use exact contact-dynamics (Posa et al. [30]), we provide a global model. In contrast to previous works that rely on approximate contact-dynamics (Todorov [35], Toussaint et al. [36]), we implement approximations that obey non-penetration and contact complementarity, fundamental properties in manipulation. The key tools to achieve this come from mixed-integer convex optimization, including piecewise McCormick envelopes and disjunctive constraints. Hencer, in this paper, we present two main contributions:

- **Modeling** of the contact-trajectory optimization problem with quasi-dynamics and polytopic objects and environments as a mixed-integer convex optimization problem. Our model retains global optimality, global infeasibility and convergence.
- **Validation** of the model on multi-contact manipulation

behaviors on planar environments, sagittal and transversal, both in simulation and on real robot environment.

Fig. 1 illustrates our vision for this model: by optimizing contact-trajectories in simple environments we can generate motion plans to complete tasks that involve complex contact interactions. We show how our model optimizes contact-trajectories for planar tasks consistently in less than 1 s.

The remainder of this paper is organized as follows: Sec. II reviews concepts and literature relevant to this work. Sec. III provides an overview of the model, its assumptions and properties. Sec. IV describes the proposed model in detail, while Sec. V discusses its implementation as a Mixed-Integer Convex Program. Sec. VI demonstrates the model with simulation and real experiments, and we conclude in Sec. VII summarizing the contributions and limitations of the work.

II. BACKGROUND

In this section we review some of the previous research most relevant to this work and introduce the concepts that we reference through the paper.

A. Trajectory Optimization

Trajectory Optimization is a popular tool to generate robot motions that need to respect some constraints [15, 20, 33]. The main benefit it provides comes from its versatility, as it generalizes between systems and problems. A significant body of work has been dedicated apply this tool to contact-rich motion generation [28, 22, 23, 30, 34, 36, 37]. Since contact is fundamentally discontinuous and dependent on geometry, modelling it requires the introduction of non-smooth constraints or discrete decision variables to determine its schedule. Hence, solution to this problem is subject to a dichotomy, either rely on fast nonlinear optimization tools [23, 30, 34], which rigorously model the problem at the cost of dependence on initialization and lack of convergence guarantees, or pose the problem as a large combinatorial optimization, which has useful theoretical properties subject to an exponential growth in complexity [28, 36, 37].

Despite the aforementioned challenges, many have successfully applied trajectory optimization through contact to locomotion problems [6]. Manipulation, however, introduces a higher dimension of complexity, as geometry and hybrid dynamics play a larger role in the generation of object motions.

B. Quasi-Dynamic Models

The idea of applying approximate physics to solve contact-rich planning problems has been pursued for many decades [25], promising a trade-off between performance and fidelity. On one extreme, some attempt to accurately model the dynamics of the entire system [30], while other completely drop the non-smooth/nonlinear aspects of the model [35] for performance. In the middle, and most relevant to us, some works approximate specific constraints of the problem while retaining other relevant aspects of contact interaction [27, 17, 36]. In the case of [27], a quasi-dynamic relaxation allows them to formulate the prehensile pushing problem with

gravity in terms of motion cones. On the other hand, [36] shows how a simplified model guided by geometry can be used to generate long-horizon trajectories involving objects and tools.

A relevant example is that of [37], where the hybrid aspects of the quadruped locomotion problem are accurately modeled, while the nonlinear relations of angular dynamics are relaxed. Our work borrows from this philosophy, where relaxations are applied to some elements of the problem, such that we can still capture the general behavior of the system and provide better performance and theoretical guarantees.

C. Mixed-Integer Programming

A Mixed-Integer Convex Program (MIP) [11] is an optimization problem of the form: find \mathbf{x}, \mathbf{z} such that:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} \in \mathcal{H}, \quad \mathbf{x} \in \mathbb{R}^{\dim(\mathbf{x})}, \quad \mathbf{z} \in \{0, 1\}^{\dim(\mathbf{z})},$$

where \mathcal{H} is a *convex* set of constraints, \mathbf{x} is a vector continuous variables and \mathbf{z} a vector of binary variables. Mixed-Integer Programs have many useful properties and applications.

In particular, will make use of two properties of these models. First, if a *convex* cost function is provided, we can always find its globally optimal solution or can prove that it is infeasible. Second, binary variables can be used to encode logical implications as linear constraints. A common technique is the big-M formulation: for a binary variable \mathbf{y} , we can incorporate logical constraints of the form $\mathbf{y} = 1 \Rightarrow A\mathbf{x} \leq b$, as the inequality $A\mathbf{x} \leq b + M(1 - \mathbf{y})$, where M is a large positive number. Through this inequality if $\mathbf{y} = 1$ then $A\mathbf{x} \leq b$ is enforced, if $\mathbf{y} = 0$ then it is not (up to M). Similar techniques include “convex-hull” and “perspective” formulations, which are better numerically conditioned than big-M formulation [24].

These properties come at the cost of exponential bounds on computation time, which grow with the number of binary variables. Through the paper, we will show how logical implications can be used to assign contacts to a surface and approximate non-convex constraints.

A significant body of research has applied Mixed-Integer Programming for trajectory optimization and model-predictive control [7, 16, 31, 32, 37].

III. APPROACH OVERVIEW

In this section we provide an overview of our model, describe its properties and discuss its assumptions. Given the trajectory of a polytopic object, this model will find a sequence of contacts interactions that achieve this motion and minimize a convex objective.

A. Inputs and Notation

Our model receives as inputs the trajectory of the object and the geometry of the object and environment. We introduce the following notation and variables:

- 1) **Object:** A polytopic rigid-body \mathcal{O} with N_v vertices and N_f facets. Each facet \mathbb{F}_f has N_v^f vertices, with nominal

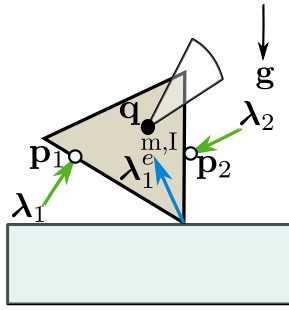


Fig. 2: For a trajectory of T time-steps, our model solves for the finger positions \mathbf{p} (dots) and contact forces $\boldsymbol{\lambda}$ (green), as well as reaction forces with the environment $\boldsymbol{\lambda}^e$ (blue) represented by separate surrogate variables.

positions \mathbf{v}_v^f , with a corresponding friction cone \mathbb{FC}_f , represented with R_d rays. In the 2D case we have $N_f = N_v$ and $N_v^f = 2$.

- 2) **Trajectory:** A set of object poses over T discrete time-steps. We describe each pose, at a time-step t , as $\mathbf{q}(t) \in \mathcal{C}$, where \mathcal{C} is the configuration space of the object. In the 2D, \mathcal{C} corresponds to the x, y, θ coordinates of $SE(2)$.
- 3) **Manipulator:** A set of N_c contact points. We describe the c_{th} contact-point, at time-step t , as $\mathbf{p}_c(t) \in \mathcal{W}$, where \mathcal{W} is the workspace. In the 2D case, the workspace simply corresponds to each position (x, y) that can be reached by the robot.
- 4) **Environment:** A polytopic environment with N_e facets, described as planes with friction cones \mathbb{FC}_e . Additionally, the free-space between the object and environment is segmented in N_R convex polytopic regions $\mathcal{R}_r = \{\mathbf{x} \in \mathcal{W} \mid A_r \mathbf{x} < b_r\}$.

A diagram describing these elements, at a fixed time-step, can be seen in Fig. 2.

B. Desired Properties

The proposed model describes the time-evolution of the multi-contact system while maintaining the following desirable properties:

- 1) **Versatility** the model must capture rigid-body (quasi) dynamics and hybrid (sticking) contact interactions while remaining agnostic to the task, object, and environment.
- 2) **Global Optimality** If there is a feasible set of contact trajectories that achieve this motion, the model will always find it. Moreover, if there is a convex cost function, the model can always be used to find the solution that globally minimizes it.
- 3) **Infeasibility Detection** If the model is infeasible, then it implies that the object motion cannot be achieved.

Throughout the upcoming sections we will discuss how the constraints of the model preserve these properties.

C. Modeling Assumptions

In order to achieve the desired properties of the model, we make the following assumptions:

- 1) Objects are rigid, with uniform contact surfaces (such that line contacts can be approximated by contact with two

vertices), and approximated as combinations of “simple” polytopes.

- 2) Object motions occur at low speeds, such that high order inertial effects are negligible.
- 3) Robot fingers have small masses, such that there is no impacts between the robot and the object.
- 4) Interactions between the object and robot are a sequence of alternating sticking contacts, such that friction cones can be approximated as R_d rays.

The upcoming sections will describe how these modeling decisions translate into a mixed-integer convex model.

IV. GLOBAL QUASI-DYNAMIC MODEL

In this section we present our global quasi-dynamic model. Each subsection will describe a set of constraints, for which we discuss their derivation and numerical implementation.

A. Force-Motion Equations

By solving the Lagrange equations for an object, assumed to be rigid, we obtain the standard force-motion equation that describes the dynamics of the system:

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \tau_g(\mathbf{q}) + H^T(\mathbf{q})\boldsymbol{\Lambda},$$

where the given object configuration is denoted by $\mathbf{q} = \begin{bmatrix} \mathbf{q}_t \\ \mathbf{q}_\theta \end{bmatrix}$, where \mathbf{q}_t corresponds to the position of the object and \mathbf{q}_θ to its orientation, $H(\mathbf{q})$ maps the effect of the contact forces $\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\lambda}^e \end{bmatrix}$ as body wrenches, τ_g describes the gravitational wrench, $M(\mathbf{q})$ is a mass matrix, and $C(\mathbf{q}, \dot{\mathbf{q}})$ maps velocities $\dot{\mathbf{q}}$ into Coriolis forces. When velocities are low and inertial effects are small, the equation of motion reduces to:

$$M(\mathbf{q})\ddot{\mathbf{q}} = \tau_g(\mathbf{q}) + H^T(\mathbf{q})\boldsymbol{\Lambda}.$$

From this form, we can decouple the equations of motion into translational and rotational components:

1) *Translational Motion:* The top rows are directly Newton’s second law applied to the center of mass of the object:

$$m\ddot{\mathbf{q}}_t(t) = m\mathbf{g} + \sum_c \boldsymbol{\lambda}_c(t) + \sum_n \boldsymbol{\lambda}_n^e(t) \quad (\text{CT1})$$

Where m is the object mass, $\boldsymbol{\lambda}_c$ is the force applied by the c_{th} finger \mathbf{p}_c , and $\boldsymbol{\lambda}_n^e$ is the environmental force applied over the n_{th} vertex of object. Eq. (CT1) is a linear sum of unknown terms, which is a convex constraint.

2) *Rotational Motion:* The bottom rows, recasting the Jacobian $H^T(\mathbf{q})$ as a sum of cross-product operations, become:

$$I\ddot{\mathbf{q}}_\theta(t) = \sum_c (\mathbf{p}_c(t) - \mathbf{q}_t(t)) \times \boldsymbol{\lambda}_c(t) + \sum_n R^r(q_\theta(t)) \mathbf{v}_n \times \boldsymbol{\lambda}_n^e(t)$$

where I is the object’s moment of inertia w.r.t its center of mass, \mathbf{p}_c is the position of the c_{th} finger, and R^r is a rotation matrix from 0 to \mathbf{q}_θ . Unfortunately, the cross-product

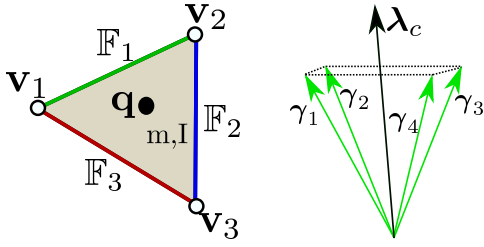


Fig. 3: (left) Contact Assignment: the object is decomposed in N_f facets and N_v vertices, and contacts forces are constrained to lie within the respective friction cone. (right) Friction cone approximation as $R_d = 4$ rays.

operation is a bilinear term, making each torque a non-convex constraint. For now, we replace each cross-product with a surrogate term and substitute the rotational dynamics with:

$$I\ddot{q}_\theta(t) = \sum_i \tau_i(t) + \sum_n R^r(\mathbf{q}_\theta(t))\mathbf{v}_n \times \boldsymbol{\lambda}_n^e(t) \quad (\text{CT2})$$

where $\tau_i \approx (\mathbf{p}_i - \mathbf{q}_t) \times \boldsymbol{\lambda}_i$ is mixed-integer convex approximation, the method by which this approximation is computed is described in detail in Sect. IV-E. Note that, fortunately, the terms corresponding to external forces are linear constraints on the external force, since $R^r(\mathbf{q}_\theta(t))$ is known a priori.

Equations (CT1) and (CT2) are often referred to as the ‘‘centroidal dynamics’’ of a system, ubiquitous in the legged locomotion community, see [6, 29].

B. Contact Scheduling

In order to ensure consistency between equations (CT1)-(CT2) and the position of the fingers, forces $\boldsymbol{\lambda}_c$ must be active only if \mathbf{p}_c is in contact with one of the facets of the object. Moreover, this contact force must be constrained to lie within its corresponding friction cone. This leads to a hybrid condition, as the constraints change depending on the position of the fingers and the shape of the object. To achieve this, we leverage the object representation as a polytope with N_f facets \mathbb{F}_f and the fingers described as points, as shown in Fig. 3 (left).

To include this constraint, at each time-step t , we introduce a binary matrix as part of the decision variables $\mathbf{T}(t) \in \{0, 1\}^{N_f \times N_c}$ that maps the position of each contact c to some facet f of the object via the constraint:

$$\mathbf{T}_{f,c}(t) = 1 \Rightarrow \begin{cases} \mathbf{p}_c(t) \in \mathbb{F}_f(t) \\ \boldsymbol{\lambda}_c(t) \in \mathbb{FC}_f(t) \end{cases} \quad (\text{CT3})$$

This constraint enforces that forces are only active when the fingers are in contacts with a facet. Since each facet \mathbb{F}_f has N_v^f vertices with position \mathbf{v}_n^f , we model the facet assignment constraint as:

$$\mathbf{p}_c(t) \in \mathbb{F}_f(t) \Leftrightarrow \mathbf{p}_c(t) = \sum_j \rho_j(t)\mathbf{v}_j^f, \quad \sum_j \rho_j(t) = 1,$$

which constrains the finger position to be a convex combination of the facet vertices, where ρ are assignment weights.

Then, we constrain the force to lie on its friction cone \mathbb{FC}_f , described with R_d rays $\gamma_{f,1}, \dots, \gamma_{f,R_d}$, with $R_d = 2$ for 2D and $R_d \geq 3$ for 3D, depicted in Fig. 3 (right), as:

$$\boldsymbol{\lambda}_c(t) \in \mathbb{FC}_f(t) \Leftrightarrow \boldsymbol{\lambda}_c(t) = \sum_k \alpha_k(t)\gamma_{f,k}, \quad \alpha_k(t) > 0,$$

which constrains each contact force to be a conic combination of the friction cone rays, where α are also assignment weights. Finally, since forces cannot be active if the fingers are not in a facet, we add the following constraint:

$$\sum_f \mathbf{T}_{f,c}(t) = 0 \Rightarrow \boldsymbol{\lambda}_c(t) = 0 \quad (\text{CT4})$$

We transcribe all \Rightarrow operator using big-M formulation. The constraints defined by Eqs. (CT3) and (CT4) are equivalent to a complementarity constraint over the contact force [30], since the contact force will only be non-zero once the finger touches the object.

1) *Alternated-Sticking*: For robustness of the optimized trajectories, we further require that finger contacts are not sliding between time-steps, as our model assumes that contact-trajectories are a sequence of sticking contacts, we model this constraint as:

$$\mathbf{T}_{f,c} = 1 \Rightarrow \mathbf{p}_c(t+1) = \sum_j \rho_j(t)\mathbf{v}_j^f, \quad (\text{CT5})$$

which enforces that if the finger is in contact at time t then it must remain sticking at time-step $t+1$ before switching to a different contact location. If we were to allow sliding-contact, we would need to represent the border of the friction cone which is a non-convex constraint [13]. This is possible, but would extend the size of the optimization problem.

C. Environment Contacts

Since we provide the object motion as an input and we assume an uniform pressure distribution on the object facets, the contact schedule between the object and the environment is also known. Hence, the model only needs to constrain reaction forces at each object vertex \mathbf{v}_n , to lie within their respective friction cone $\mathbb{FC}_n^e(t)$, with R_d rays $\gamma_{n,1}^e(t), \dots, \gamma_{n,R_d}^e(t)$. This constraint is imposed, for the n th vertex at time-step t , as:

$$\boldsymbol{\lambda}_n^e(t) \in \mathbb{FC}_n^e(t) \quad (\text{CT6})$$

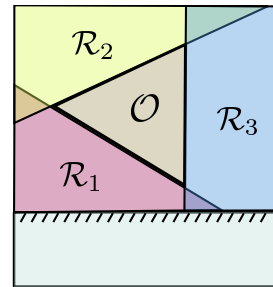


Fig. 4: Non-Penetration Constraint: The free-space between the object and the environment is decomposed in N_R convex regions \mathcal{R} , including the facets of the object.

As before, the friction cone constraint is included as $\lambda_n^e(t) = \sum_k \alpha_k^e(t) \gamma_n^e(t)$, $\alpha_k^e(t) \geq 0$, where α^e are assignment weights for each ray. Since these vertex contacts can also slide, some of the assignment weights must be fixed to zero to have the force in the border of the friction cone. This can be done when setting-up the optimization problem, since the contact modes between each vertex and the environment are known from the given trajectory.

D. Non-Penetration

Naturally, fingers cannot penetrate the object nor the environment. A strategy to enforce this is to segment the free-space into N_R convex, possibly overlapping, regions \mathcal{R}_r that cover the free space of the workspace [31, 21, 37], as shown in Fig. 4. Each finger is then constrained to lie within one of these regions. This constraint is added through a binary decision matrix $\mathbf{R}(t) \in \{0, 1\}^{N_R \times N_c}$, such that:

$$\mathbf{R}_{r,c}(t) = 1 \Rightarrow \mathbf{p}_c(t) \in \mathcal{R}_r(t) \quad (\text{CT7})$$

with $\sum_r \mathbf{R}_{r,c}(t) = 1$, $\forall c$. Here, region assignment is done as:

$$\mathbf{p}_c(t) \in \mathcal{R}_r(t) \Leftrightarrow A_r(t) \mathbf{p}_c(t) < b_r(t)$$

which are all linear constraints on the finger positions. Finding these regions is a separate optimization problem, examples include [8].

E. Modeling Approximations via McCormick Envelopes

The constraints defined by Eq. CT2 include a cross-product operation, which is a non-convex function due to the presence of bilinear equalities. For the purpose of this paper, we refer to a bilinear equality constraint as a relation of the type:

$$w = u \cdot v$$

where u , v and w are decision variables. Concretely, each cross-product τ adds $4|\tau| - 6$ bilinear equalities to the model. To illustrate their non-convexity, we plot the surface $w = uv$ in Fig. 5a. While there are several methods to approximate or relax non-convex constraints of this type, we are interested in

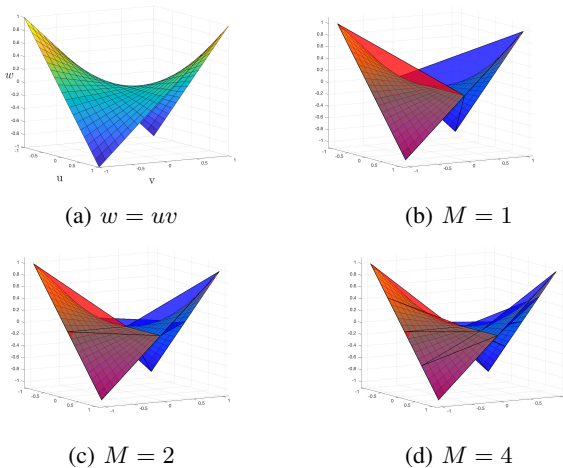


Fig. 5: (a) Bilinear equality curve for $w = uv$, (b)-(d) Piecewise McCormick Envelopes with different levels of accuracy.

an approximation that 1) can be embedded in a global model and 2) preserves the hybrid structure of contact. A technique that achieves this purpose is that of *Piecewise McCormick Envelopes*. Initially proposed by McCormick in [26], this approximation covers the bilinear surface $w = uv$ with M convex envelopes, each being the convex-hull of the surface between a segment on the uniform set $-u_{M/2}, \dots, u_{M/2}$ and $v \geq 0, v \leq 0$. Examples of these envelopes are shown in Fig. 5b, blue for $v \geq 0$ and red for $v \leq 0$. Then, each approximation of w is constrained to lie within the envelopes with a binary decision matrix $\mathbf{W}(t) \in \{0, 1\}^{2 \times M}$ as:

$$\mathbf{W}_{1,k}(t) = 1 \Rightarrow \begin{cases} w \geq u_{k-1}v \\ w \geq u_k v + u - u_{k-1} \\ w \leq u_{k-1}v + u - u_k \\ w \leq u_k v \\ u_{k-1} \leq w \leq u_k \\ v \geq 0 \end{cases} \quad (\text{CT8})$$

$$\mathbf{W}_{2,k}(t) = 1 \Rightarrow \begin{cases} w \leq u_{k-1}v \\ w \leq u_k v - u - u_{k-1} \\ w \geq u_{k-1}v - u - u_k \\ w \geq u_k v \\ u_{k-1} \geq w \geq u_k \\ v \leq 0 \end{cases}, \quad (\text{CT9})$$

which constraint w to lie within one of the envelopes, depending on the sign of v . This approximation is not exact; however, it provides several useful properties:

- 1) The segmentation provides upper and lower bound on the quality of the approximation.
- 2) A larger M makes the approximation arbitrarily tight, at the cost of a larger binary matrix.
- 3) The approximation preserves the hybrid and nonlinear structure of the bilinear surface, as:

$$\begin{aligned} u \cdot v \geq 0 &\Rightarrow w \geq 0 \\ u \cdot v \leq 0 &\Rightarrow w \leq 0 \\ u \cdot v = 0 &\Rightarrow w = 0 \end{aligned}$$

- 4) If the relaxed envelope constraint is infeasible then the original bilinear constraint is also infeasible, as the envelopes cover the original curve in their solution space.

Examples of approximations of different sizes are shown in Fig. 5a-5d.

V. CONTACT-TRAJECTORY OPTIMIZATION

From the constraints presented above, we formulate a Mixed-Integer Optimization problem. We summarize all the decision variables required for our model in Table I. For notation convenience we define three sets of decision variables: contact-trajectories $\mathcal{X} = \{\mathbf{p}, \lambda, \lambda^e, \tau\}$, assignment weights $\mathcal{Y} = \{\rho, \alpha, \alpha^e\}$, and binary matrices $\mathcal{T} = \{\mathbf{T}, \mathbf{R}, \mathbf{W}\}$.

Name	Description	Size	C/B
\mathbf{p}	Point Locations	$D \times N_c \times T$	C
$\boldsymbol{\lambda}$	Contact Forces	$D \times N_c \times T$	C
$\boldsymbol{\lambda}^e$	External Forces	$D \times N_v \times T$	C
$\boldsymbol{\tau}$	Torque Approximation	$(4 \boldsymbol{\tau} - 6) \times N_c \times T$	C
ρ	Facet weights	$\sum_f N_v^f \times N_c \times T$	C
α	Friction Cone Weights	$R_d \times N_c \times T$	C
α^e	External Cone Weights	$R_d \times N_v \times T$	C
\mathbf{T}	Contact Assignment	$N_f \times N_c \times T$	B
\mathbf{R}	Non-Penetration	$N_R \times N_c \times T$	B
\mathbf{W}	McCormick Envelope	$2M \times (4 \boldsymbol{\tau} - 6) \times N_c \times T$	B

TABLE I: Summary of Decision Variables (C: Continuous, B: Binary).

A. Optimization problem

Adding the constraints and all the decision variables, we transcribe the optimization problem into **MIQP1**:

$$\mathbf{MIQP1} : \underset{\mathcal{X}, \mathcal{Y}, \mathcal{T}}{\text{minimize}} J = [\mathcal{X} \quad \mathcal{Y} \quad \mathcal{T}] Q \begin{bmatrix} \mathcal{X} \\ \mathcal{Y} \\ \mathcal{T} \end{bmatrix} + q^T \begin{bmatrix} \mathcal{X} \\ \mathcal{Y} \\ \mathcal{T} \end{bmatrix}$$

subject to:

- 1) For time-step $t = 1$ to $t = T$:
 - a) Quasi-Dynamics (CT1)-(CT2).
 - b) For fingers $c = 1$ to $c = N_c$
 - Contact-Trajectory Assignment (CT3)-(CT5).
 - Non-Penetration (CT7).
 - c) Environmental Contact (CT6).
 - Pre-fix weights α^e when sliding.
 - d) For bilinear terms $b = 1$ to $b = N_c(4D - 6)$:
 - Piecewise McCormick Envelope (CT8)-(CT9).

Where Q is a positive-semi-definite (PSD) square matrix and q is a column vector of appropriate size, these matrices can be chosen according to the problem.

B. Properties of the Model

The formulation of the problem can be categorized as a Mixed-Integer Quadratic Program (MIQP); this type of problem has several useful properties [11]. Mainly, if given sufficient time (with worst-case exponential complexity), a solver can always find the global solution to the optimization problem. This also implies that it does not require any form of initialization or warm-start. Finally, If **MIQP1** results infeasible we can guarantee that the original problem, with exact bilinear equality constraints, is also infeasible [26].

The complexity of the program grows exponentially with the number of binary variables in the model, defined by the number of fingers, tightness of the McCormick envelopes, shape of the object and environment.

VI. VALIDATION AND APPLICATIONS

To demonstrate some of the capabilities of our model, we implement **MIQP1** and assess its application for set of traditional manipulation problems. First, we aim to validate the model’s ability to optimize simple manipulation behaviors and detect infeasible ones. We then show different applications of

our model for manipulation problems that involve interaction with the robot and the environment. Finally, we execute a set of open-loop experiments to illustrate how this model transfers to a real-world set-up.

We generate all the trajectories in MATLAB R2019b, running on an Intel Core i9 laptop with Mac OS X High Sierra. We use Gurobi 8.1.0 [12], an off-the-shelf optimization software, as our MIP solver. All of our tests are done in two-dimensional set-ups, such that $|\boldsymbol{\tau}| = |\mathbf{p}| = N_v^f = R_d = 2$. We fix the accuracy of each piecewise McCormick envelopes to $M = 4$. We manually generate each object trajectory and segment the free-space of each task into convex regions \mathcal{R} . For all problems, we add a quadratic cost-function that minimizes the applied force and smooths the finger trajectories:

$$J = \sum_{t=1}^T \sum_{c=1}^{N_c} \|\boldsymbol{\lambda}_c(t)\|^2 + \|\ddot{\mathbf{p}}_c(t)\|^2 - \beta_c(t)$$

Second derivatives are computed within the model with backwards-Euler scheme for simplicity and numerical stability. The term β is a lower-bound convex-approximation of the distance between each contact-force and the border of its friction cone, computed as in [2, 17], this term has to be linear in order for the cost-function to be convex.

A. Model Validation

We start by validating the functionality of the model in several simple manipulation problems. In particular, we show its ability to optimize contact-trajectories for “primitive” object motions and to detect when an object motion is infeasible. We generate a set of object trajectories in the sagittal $-XZ$ -plane for which the optimal solution can be intuitively found. For this, we use three objects: 1) a block, 2) a triangle, a 3) a non-convex object. Unless otherwise specified, all surfaces have a friction coefficient of $\mu = 0.1$. Then, we generate the following trajectories of $T = 5$ time-steps:

Block 1-finger sliding (Fig. 6a), 2-finger pivoting (Fig. 6b), 2-finger grasping (Fig. 6c). We observe that one finger is sufficient to slide the object by pushing it. Grasping directly from the ground requires two or more contacts in order to lift the object. In the case of pivoting, a surface with $\mu = 0.1$ is not sufficient for 1-finger pivoting and the solver only finds a solution when $N_c \geq 2$, which allows to create internal force to increase friction.

Triangle 2-finger pivoting (Fig. 6f). Consistent with physical intuition, $\mu = 0.1$ does not provide enough friction for the object to pivot without sliding; hence, the model chooses to place a second finger to push on the ground a generate additional reaction force, providing enough torque. Similar to the block case, fixing $N_c = 1$ leads the solver to report infeasibility.

Non-Convex Object 1-finger sliding (Fig. 6d) and 1-finger pivoting (Fig. 6e). For both tasks, one finger is sufficient. In contrast to the other objects, the non-convex “end” of the tool can be used to generate sufficient torque and pivot with only one finger.

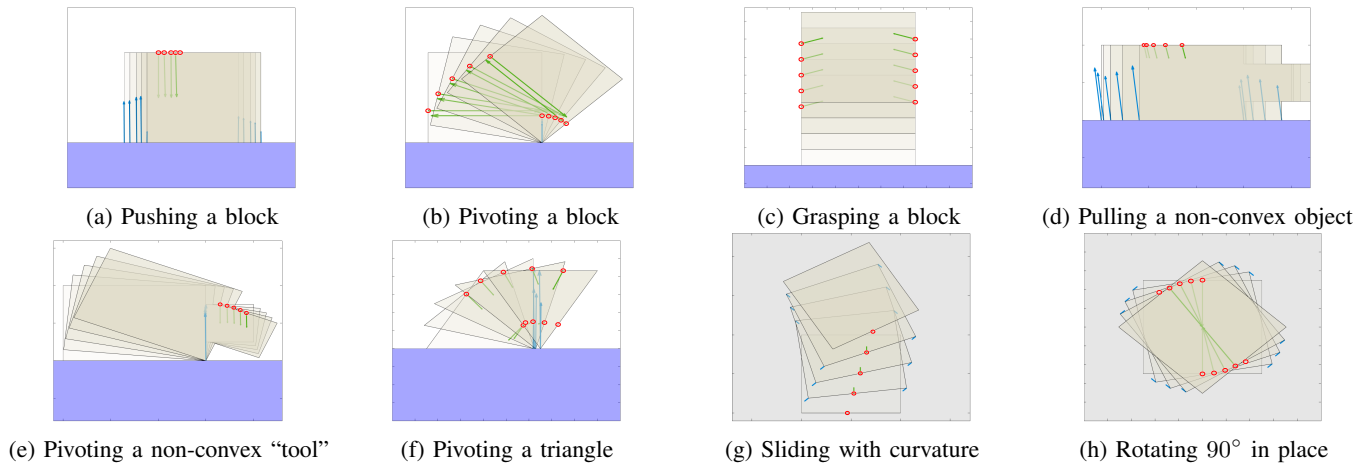


Fig. 6: Primitive behaviors optimized with our model, without any initialization or seeding. Red points are the finger locations, green arrows the applied forces, and blue arrows represent the environmental contacts.

Transverse manipulation 1-finger curved-pushing (Fig. 6g) and 2-finger in-place rotation (Fig. 6h). For straight line pushing or pushing with small curvature, one finger is sufficient, as it can generate enough torque to slide and rotate. However, two fingers are required to rotate the block around its geometric center.

As a reference, all trajectories are optimized in the range of 0.04s to 0.44s of computation. We stress the benefits of global optimality in this problem, since small changes in the friction coefficient or geometry of the object lead to different solutions. Furthermore, the ability to report when a task is infeasible, either because it requires more than one finger or because it is physically impossible, also provides useful insight on the primitive itself.

B. Applications

Once we have proven that our framework can effectively generate simple behaviors, we aim to show how it can also reason about longer horizon tasks that involve complex contact interactions. For this, we show two examples of “extrinsic” dexterity [5], where contacts with the environment are essential parts of the manipulation problem, on two different set-ups with $T = 10$ time-steps. In both cases, the object trajectories were specified manually on a set-up with a block and a wall. We optimize these “extrinsic” re-orienting strategies in two separate planes:

Sagittal Accounting for the effect of gravity, the object is pushed to a wall, pivoted w.r.t. it, and lifted vertically. The execution of this motion is shown in Fig. 8a. The optimal solution is computed in under 10 sec.

Transverse In this case, the object slides against a wall and is then pivoted with respect to it. This trajectory is shown in Fig. 8b. The optimal solution finds a sticking contact location for each finger. The problem is solved in 2 sec.

C. Experimental Validation

In order to demonstrate that the contact-trajectories generated by this model translate to the real world, we conduct

experiments on a two-arm robot manipulating and a cubic object on a table.

Our robotic platform is an ABB YuMi[®] (IRB-14000) robot, which has two 7 DOF arms with a custom point-finger attached to each end-effector. We work with a Robot Operating System (ROS) setup, interfaced with MATLAB R2019, and run all the demonstrations in an open-loop fashion, guided through position commands.

Transverse Manipulation Our first demonstration focuses on planar sliding on a uniform surface, replicating the behaviors in Figs. 6g, 6h, and 8b. Snapshots of the execution of each task are shown in Fig. 7 (top to middle). Despite the open-loop fashion of these demonstrations, we achieve a reliable execution of the optimized contact-trajectory, as long as the trajectory is executed slow enough, otherwise impacts occur and inertial effects are noticeable.

Sagittal Manipulation Our second demonstration shows how our model can accurately generate motions that interact with gravity. For this, we replicate the pivoting and grasping behaviors, depicted in Figs. 6b and 6c. Executions are shown in Fig. 7 (bottom two). In this example, however, trajectories are brittle and highly-dependent on the initial pose of the object and the robot contacts. This points at the importance of tracking position along with forces, as contacts must remain sticking in order to execute these tasks correctly.

In both cases, accuracy is achieved by placing the object in the precise initial condition. The general scenario, with uncertainty, requires a controller and perception to execute each motion.

VII. DISCUSSION

In this paper we have presented a global model for the *Contact-Trajectory Optimization* problem on polytopic objects. By applying quasi-dynamic approximations, relaxing high-order coriolis effects and applying McCormick envelopes on the non-convex relations, we obtain a set of constraints that provide optimality, infeasibility, and convergence guarantees.

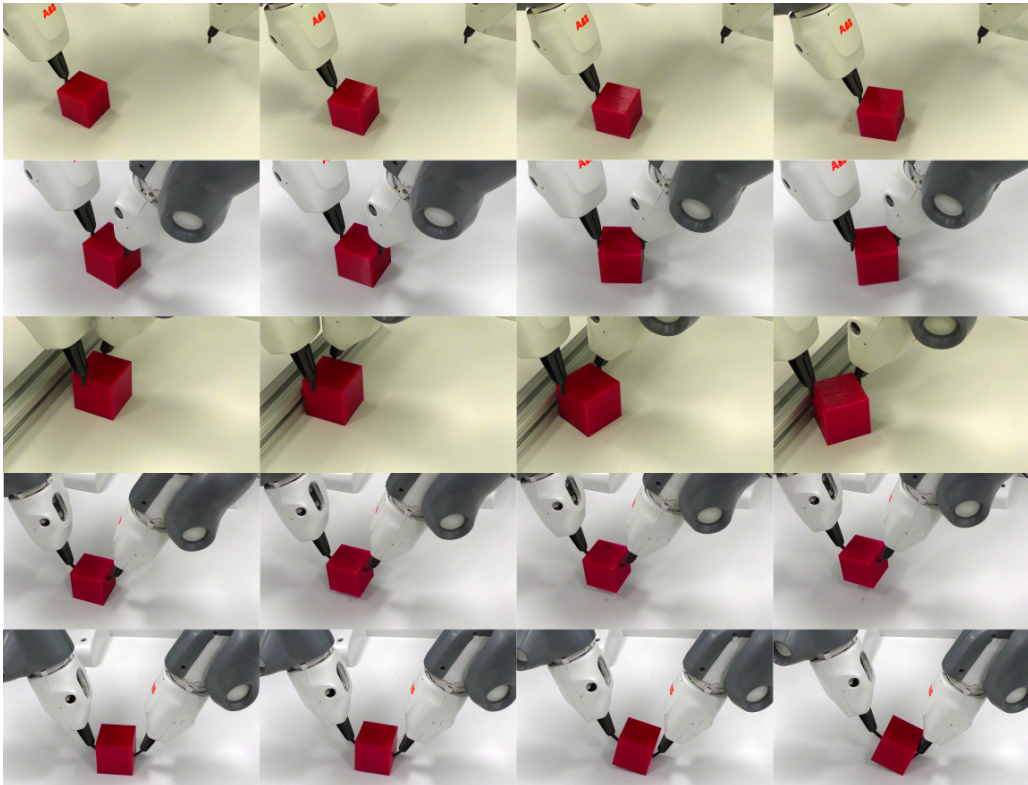
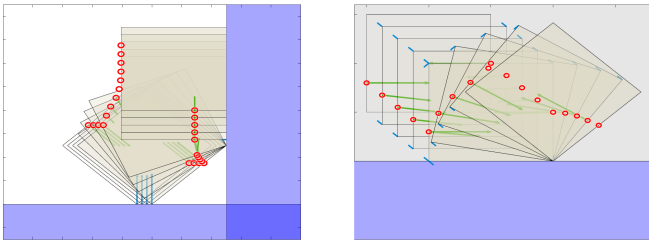


Fig. 7: **Experimental validation of our model** Our optimized behaviors can be applied for real world execution, snapshots of each open-loop experiment are shown (each blue box corresponds approximately to the goal pose of the object). Top to bottom: 1) transversal pushing with a desired angle. 2) Rotating 45° in-place. 3) Transversal pivoting against a wall. 4) Grasping vertically. 5) Sagittal pivoting.



(a) Pivoting against a wall with gravity

(b) Planar sliding with two fingers and a wall

Fig. 8: Two applications of our model for extrinsic dexterity in the sagittal and transverse planes.

Our model accounts for the environment and object shape, thus being applicable to multiple manipulation problems.

We implement our model as Mixed-Integer optimization program and demonstrate its ability to optimize simple manipulation primitives and contact interactions between the object, robot and environment. Trajectories are found between 0.04 s to 0.44 s, using off-the-shelf optimization software. Our long term vision is that the execution of these plans will be supported by a low level controller that uses tactile-feedback to enforce contact conditions, as initially explored in [17].

Model Limitations Perhaps the main limitation of this approach comes from needing to specify as input the object motion. While in many cases object trajectories are intuitive to specify, many simple motions lead to infeasible contact-trajectories, such a lifting a steep triangle.

The second main limitation is the restriction to alternating sticking-contacts. This is not necessarily a limitation in the 2D case since the sliding condition can be included through a binary decision matrix that encodes the contact mode of each finger at every time-step [16]. The 3D case is more troubling, since the friction cone border constraints are non-convex and these would have to be approximated in some way [13].

Future Work The first set of extensions to this work arise from its main limitations: incorporating sliding contact, as described above, and mitigating the dependence on an provided trajectory. A potential way to achieve the later would be to plan for the motion in a sampling-based fashion, applying this model in an inverse-dynamics fashion similar to [4]. Further validation in more complex 3D tasks [18] is also important.

Since our model outputs trajectories with position and force, a natural extension would be to combine this model with a state-of-the-art feedback controller [3, 28, 14, 16, 32]. This is particularly relevant under recent advances on localized tactile sensing [10, 17, 9], which could allow us to generate more dynamic motions under uncertainty.

Finally, we are interested in studying potential applications of this model in the context of learning manipulation, either in the contexts of imitation learning or affordances for contact-rich manipulation [38], and optimizing robust manipulation primitives, which can naturally funnel uncertainty [1, 19].

REFERENCES

- [1] B. Aceituno-Cabezas, J. Ballester, and A. Rodriguez. Certified grasping. In *ISRR*, 2019.
- [2] Bernardo Aceituno-Cabezas, Carlos Mastalli, Hongkai Dai, Michele Focchi, Andreea Radulescu, Darwin G Caldwell, José Cappelletto, Juan C Grieco, Gerardo Fernández-López, and Claudio Semini. Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization. *RA-L and ICRA*, 2018.
- [3] Alp Aydinoglu, Victor Preciado, and Michael Posa. Contact-aware controller design for complementarity systems. In *To appear in ICRA*, 2020.
- [4] N. Chavan-Dafle and A. Rodriguez. Stable prehensile pushing: In-hand manipulation with alternating sticking contacts. In *ICRA*, 2018.
- [5] Nikhil Chavan Dafle, Alberto Rodriguez, Robert Paolini, Bwei Tang, Siddhartha S Srinivasa, Michael Erdmann, Matthew T Mason, Ivan Lundberg, Harald Staab, and Thomas Fuhlbrigge. Extrinsic dexterity: In-hand manipulation with external forces. In *ICRA*. IEEE, 2014.
- [6] Hongkai Dai, Andrés Valenzuela, and Russ Tedrake. Whole-body motion planning with centroidal dynamics and full kinematics. In *Humanoids*. IEEE, 2014.
- [7] Robin Deits and Russ Tedrake. Footstep planning on uneven terrain with mixed-integer convex optimization. In *2014 IEEE-RAS international conference on humanoid robots*, pages 279–286. IEEE, 2014.
- [8] Robin Deits and Russ Tedrake. Computing large convex regions of obstacle-free space through semidefinite programming. In *WAFR*. Springer, 2015.
- [9] L. Donghyun J. Romero E. Chuah, L. Epstein and S. Kim. Bi-modal hemispherical sensor: A unifying solution for three axis force and contact angle measurement. In *IROS*, 2019.
- [10] M.Liu J. Li E. Adelson E. Donlon, S. Dong and A. Rodriguez. Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger. In *IROS*, 2018.
- [11] Christodoulos A Floudas. *Nonlinear and mixed-integer optimization: fundamentals and applications*. Oxford University Press, 1995.
- [12] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2018. URL <http://www.gurobi.com>.
- [13] Maximilian Haas-Heger, Garud Iyengar, and Matei Ciocarlie. Passive reaction analysis for grasp stability. *T-ASE*, 2018.
- [14] Weiqiao Han and Russ Tedrake. Local trajectory stabilization for dexterous manipulation via piecewise affine approximations. *arXiv preprint arXiv:1909.08045*, 2019.
- [15] Charles R Hargraves and Stephen W Paris. Direct trajectory optimization using nonlinear programming and collocation. *Journal of guidance, control, and dynamics*, 1987.
- [16] F. Hogan and A. Rodriguez. Feedback control of the pusher-slider system: A story of hybrid and underactuated contact dynamics. In *WAFR*, 2016.
- [17] F. Hogan, J. Ballester, S. Dong, and A. Rodriguez. Tactile dexterity: Manipulation primitives with tactile feedback. In *ICRA*, 2020.
- [18] Yifan Hou, Zhenzhong Jia, and Matthew T Mason. Fast planning for 3d any-pose-reorienting using pivoting. In *ICRA*. IEEE, 2018.
- [19] Aaron M Johnson, Jennifer E King, and Siddhartha Srinivasa. Convergent planning. *RA-L*, 2016.
- [20] Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. Stomp: Stochastic trajectory optimization for motion planning. In *ICRA*. IEEE, 2011.
- [21] Scott Kuindersma, Robin Deits, Maurice Fallon, Andrés Valenzuela, Hongkai Dai, Frank Permenter, Twan Koolen, Pat Marion, and Russ Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous robots*, 2016.
- [22] C. Karen Liu. Dextrous manipulation from a grasping pose. In *SIGGRAPH*, 2009.
- [23] Zachary Manchester and Scott Kuindersma. Variational contact-implicit trajectory optimization. In *ISRR*. Springer, 2017.
- [24] Tobia Marcucci and Russ Tedrake. Mixed-integer formulations for optimal control of piecewise-affine systems. In *HSCC*, 2019.
- [25] Matthew T Mason. *Mechanics of robotic manipulation*. 2001.
- [26] Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems. *Mathematical programming*, 1976.
- [27] R. Holladay N. Chavan-Dafle and A. Rodriguez. Planar in-hand manipulation via motion cones. In *IJRR*, 2018.
- [28] F. Hogan N. Doshi and A. Rodriguez. Hybrid differential dynamic programming for planar manipulation primitives. In *ICRA*, 2020.
- [29] Brahayam Ponton, Alexander Herzog, Andrea Del Prete, Stefan Schaal, and Ludovic Righetti. On time optimization of centroidal momentum dynamics. In *ICRA*. IEEE, 2018.
- [30] Michael Posa, Cecilia Cantu, and Russ Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *IJRR*, 2014.
- [31] Arthur Richards and Jonathan How. Mixed-integer programming for control. In *ACC*. IEEE, 2005.
- [32] Sadra Sadraddini and Russ Tedrake. Sampling-based polytopic trees for approximate optimal control of piecewise affine systems. In *ICRA*. IEEE, 2019.
- [33] John Schulman, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, and Pieter Abbeel. Finding locally optimal, collision-free trajectories with sequential convex optimization. In *RSS*, 2013.
- [34] Jean-Pierre Sleiman, Jan Carius, Ruben Grandia, Martin Wermelinger, and Marco Hutter. Contact-implicit trajectory optimization for dynamic object manipulation. In

IROS, 2019.

- [35] Emanuel Todorov. A convex, smooth and invertible contact model for trajectory optimization. In *ICRA*. IEEE, 2011.
- [36] Marc Toussaint, Kelsey R Allen, Kevin A Smith, and Joshua B Tenenbaum. Differentiable physics and stable modes for tool-use and manipulation planning. In *RSS*, 2018.
- [37] Andrés Klee Valenzuela. *Mixed-integer convex optimization for planning aggressive motions of legged robots over rough terrain*. PhD thesis, Massachusetts Institute of Technology, 2016.
- [38] A. Zeng, S. Song, K.T. Yu, and et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. *IJRR*, 2019.