Robotics: Science and Systems 2024
Delft, Netherlands, July 15-July 19, 2024

Towards Tight Convex Relaxations for Contact-Rich Manipulation

Bernhard Paus Graesdal[1], Shao Yuan Chew Chia[2], Tobia Marcucci[1],
Savva Morozov[1], Alexandre Amice[1], Pablo A. Parrilo[1], and Russ Tedrake[1]

***Abstract*— We present a novel method for global motion planning of robotic systems that interact with the environment through contacts. Our method directly handles the hybrid nature of such tasks using tools from convex optimization. We formulate the motion-planning problem as a shortest-path problem in a graph of convex sets, where a path in the graph corresponds to a contact sequence and a convex set models the quasi-static dynamics within a fixed contact mode. For each contact mode, we use semidefinite programming to relax the nonconvex dynamics that results from the simultaneous optimization of the object's pose, contact locations, and contact forces. The result is a tight convex relaxation of the overall planning problem, that can be efficiently solved and quickly rounded to find a feasible contact-rich trajectory. As an initial application for evaluating our method, we apply it on the task of planar pushing. Exhaustive experiments show that our convex-optimization method generates plans that are consistently within a small percentage of the global optimum, without relying on an initial guess, and that our method succeeds in finding trajectories where a state-of-the-art baseline for contact-rich planning usually fails. We demonstrate the quality of these plans on a real robotic system.**

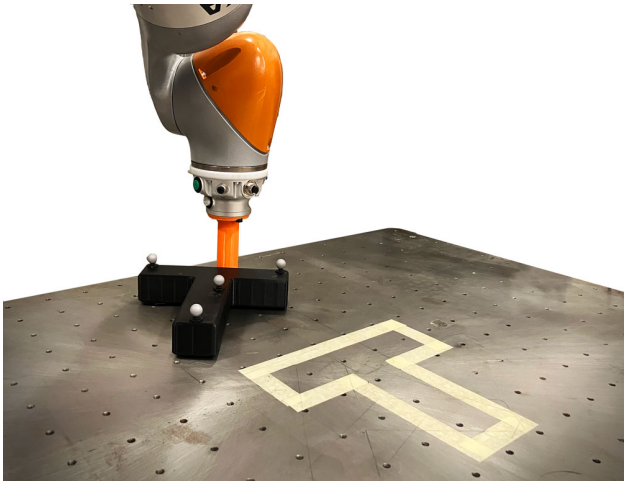


Fig. 1: The experimental planar pushing setup. A cylindrical finger is attached to a robotic arm that is pushing a T-shaped object on the table into its target configuration.

I. Introduction

Optimal planning and control through contact is an important and challenging problem in robotics, which includes tasks like robot locomotion and dexterous manipulation. It generally involves both a hybrid and underactuated dynamical system, making planning and control difficult. A variety of methods have been proposed to tackle this problem. Many of them artificially decouple the discrete and the continuous components of the problem, yielding suboptimal trajectories that do not take full advantage of the rich contact dynamics. Approaches that blend the discrete and continuous components often do so locally (around a given trajectory) and are unable to reason about the global problem; or rely on expensive global optimization algorithms that scale poorly.

In this work, we introduce a method that naturally blends the discrete logic and the continuous dynamics of planning through contact into a convex optimization problem. Specifically, we formulate the planning problem as a Shortest Path Problem (SPP) in a Graph of Convex Sets (GCS), a class of mixed discrete/continuous optimization problems that can be effectively solved to global optimality [1]. Paths in this graph correspond to different contact sequences, and the convex sets model the quasi-static contact dynamics within a fixed contact mode. These quasi-static dynamics are bilinear (therefore nonconvex), since we simultaneously optimize over the object pose, contact locations, and contact forces. Our method approximates these bilinearities using a tight Semidefinite Programming (SDP) relaxation for each contact mode. The methods from [1] are then leveraged to produce a tight convex relaxation of the global planning-through-contact problem. This relaxation can be quickly solved and rounded to obtain a feasible contact-rich trajectory. By comparing the cost of the rounded solution and the convex relaxation, our motion planner also provides us with tight optimality bounds for the trajectories that it designs.

As a first application for evaluating our method, this work explores the task of planar pushing, first studied by Mason in [2]. Planar pushing has applications that span from warehouse automation to service robotics, and although it is among the simplest examples of non-prehensile manipulation, current state-of-the-art approaches are still unable to solve this problem reliably to global optimality. The method proposed in this paper can simultaneously reason about the high-level discrete mode switches and the low-level continuous dynamics of planar pushing in a global fashion. We evaluate our motion planner through thorough numerical experiments, which show that the trajectories we generate typically have a very small optimality gap (10% on average). We also demonstrate our approach on the real robotic system shown in in Figure 1. We emphasize that, although we study the tightness of our relaxations on the particular domain

[1] Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA. E-mail: {graesdal, tobiam, savva, parrilo, russt}@mit.edu
[2] Department of Computer Science, Harvard University, Cambridge MA, USA. E-mail: shaoyuan_chewchia@college.harvard.edu

of planar pushing, the technique we introduce generalizes naturally to more complex multi-contact problems.

II. Related Work

In this section we review the multiple approaches that have been proposed in the literature to solve planning problems involving contacts. We categorize the existing approaches into learning-based, sampling, local and global optimization.

The learning methods that have been proven most effective for contact-rich tasks are Reinforcement Learning (RL) and Behavior Cloning (BC). RL has proven particularly effective due to its ability to bypass the low-level combinatorial contact decisions by using stochastic approximation of the system's dynamics [3], [4], [5]. In particular, RL has shown impressive robustness for real-world quadruped locomotion [6], and has been successfully applied to manipulation tasks in simulation [7], [8], real-world robotic manipulation [9], [10], and more recently on real-world dexterous hands [11], [12], [13], [14]. Similarly, BC methods have been successfully applied to many real-world manipulation tasks [15], [16], [17], [18], including planar pushing [15], [19]. Although RL and BC are applicable to a variety of systems, RL methods tend to require expensive training and task-specific user-shaped reward functions, while BC requires costly expert demonstrations that are typically generated by human teleoperation. The method we propose in this paper can potentially be used to automatically generate large sets of expert demonstrations of very high quality for a variety of different tasks.

Sampling-based methods have also proven effective for planning contact-rich trajectories. These algorithms often alternate between a high-level discrete search and a low-level continuous trajectory optimization. Although, in principle, this facilitates some level of global reasoning, the inability to optimize the discrete and continuous decisions jointly makes it hard to find globally optimal solutions, or even quantify the suboptimality of the solutions found. Another key challenge is that the feasible robot configurations lie on lower-dimensional contact manifolds, necessitating guided exploration in sampling. Existing methods either assume the ability to sample from individual contact modes [20], [21], [22], [23], guide the search by estimating reachable sets in configuration space [24], [25], or combine smoothed contact models with task-specific contact samplers [26]. The necessity for hand-crafted samplers, the lack of global reasoning, and the uncertainty around the plan quality often makes sampling-based contact planners difficult to use in practice.

Local optimization methods typically use contact-implicit models, where the contact is modelled as the solution set of complementarity constraints. These methods are amenable to continuous optimization by either producing smooth approximations to the contact dynamics [27], [28], [29], [30], [31], or directly operating on the complementarity formulation [32], [33], [34], [35], [36]. While these methods have shown great results in both locomotion and manipulation, they are inherently local and require high-quality initial guesses, which can be difficult to obtain [37].

Global optimization methods for planning through contact typically use Mixed-Integer Programming (MIP) to jointly optimize over discrete contact modes and continuous robot motions. A common approach is to produce a Piecewise Affine (PWA) approximation of the nonlinear dynamics, casting the problem as an MIP [38]. This and similar approaches have been applied to push-recovery [39], locomotion [40], and feedback control for planar pushing [41], [42]. Recently, efficient task-specific PWA approximation have also been obtained by pruning large neural networks [43]. The main limitation of these approaches is the MIP runtime, which can grow exponentially with the number of integer variables. In addition, PWA models are also unable to accurately capture the smooth contact dynamics within a fixed contact mode. At the cost of a further slowing down computations, the latter issue can be tackled by introducing additional binary variables and approximating the nonconvexities with piecewise linear functions (see, e.g., [44] or [45]), possibly in combination with tailored convex relaxations [46]. Alternatively, general Mixed-Integer Nonconvex Programming (MINCP) approaches have been used in robotics for momentum-based footstep planning [47], but their applicability is limited to very small problem instances.

Our method is similar to the ones in the last category, but is substantially more efficient. Rather than resorting to MINCP or piecewise linear approximations combined with mixed-integer programming, we depart from prior work by leveraging a tight convex relaxation for the bilinearities in each contact mode while also leveraging the methods from [1] to produce a tight convex relaxation for the global planning-through-contact problem. The tightness of both relaxations enables us to obtain near-globally optimal solutions by solving the entire problem as a single SDP, followed by a quick rounding step. This is the first work to our knowledge that is able to effectively solve the global planning-through-contact problem as a single convex program.

III. Problem Statement

As a first application of our method, we explore planar pushing, a non-prehensile manipulation task where the robot uses a cylindrical finger to manipulate the motion of an object resting on a flat surface. We shall refer to the manipulated object as the *slider*, the robot finger as the *pusher*, and to the overall system as the *slider-pusher* system. We assume that the slider and the pusher are rigid bodies, and restrict the problem to planar motion, such that the slider is always making contact with the underlying surface. We also assume that the slider is a prism-shaped object with known geometry, which can be nonconvex, and a known (but arbitrary) COM. See Figure 1 for a picture of the the real-world problem setup.

As the slider is pushed, it is subject to a contact wrench from the pusher and a friction wrench from the underlying surface (we assume a uniform pressure distribution between the slider and the surface). We assume quasi-static dynamics

(i.e., no work is done by impacts) and that the velocities are sufficiently low to make inertial forces negligible. Under this assumption, for the slider to move, the contact wrench applied by the pusher must balance the friction wrench from the underlying surface. We assume isotropic Coulomb friction, i.e., that the coefficient of friction is constant, and the friction force at every contact point must have a constant magnitude and oppose the direction of motion during sliding. Although these assumptions are common in the literature on non-prehensile modelling [48], [49] and planning [50], [51], we note that a more complex friction model could in principle be incorporated directly into our framework or as a post-processing step.

The described slider-pusher system is a hybrid, underactuated system. The problem is underactuated because the slider cannot move on its own, and because Coloumb friction limits the range of contact forces that can be applied by the pusher. Furthermore, the problem has hybrid dynamics, as there can be either no contact between the slider and pusher, or the contact can be sticking, sliding left, or sliding right relative to the contact point.

IV. High-Level Approach

The first step in formulating our motion planning method is to consider the dynamics and kinematics in a fixed contact mode. By formulating the manipulation problem in the task space, we are able to describe the system with only quadratic and linear constraints. More specifically, quadratic equality constraints arise from considering elements of $\mathrm{SO}(2)$, rotations of velocities and forces to the world frame, and the contact torque as the cross-product between the contact arm and force. We thus formulate the motion planning problem within a specific contact mode as a nonconvex Quadratically Constrained Quadratic Program (QCQP), which we approximate as a convex program using a semidefinite relaxation.

The second step in our method is to formulate the global motion planning problem as an SPP in a GCS [1]. For each of the contact modes in the slider-pusher system we add its associated convex program as a vertex to the graph, meaning that the feasible set of trajectories is added as a convex set and the cost is added as the vertex cost. Additionally, we decompose the collision-free subset of the configuration space into convex regions, which are also added to the graph to encode collision-free motion planning, similar to [52]. By representing the desired initial and target configurations with vertices in the graph, the globally optimal solution to the motion planning problem now corresponds mathematically to the shortest path from the source to the target in the graph.

The techniques from [1] allow us to write a convex relaxation of this planning problem which is itself an SDP that can be solved efficiently. The solution of this relaxation is then rounded to obtain a feasible solution to the motion planning problem, where the optimality gap can be bounded by comparing the cost of the feasible solution and the relaxation.

V. Background and Optimization Tools

A. Shortest paths in graph of convex sets

In this section, we briefly review the formulation for the SPP in a GCS. For further details, the reader is referred to [1]. We define a *Graph of Convex Sets (GCS)* as a directed graph $G = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E} \subset \mathcal{V}^2$, where each vertex $v \in \mathcal{V}$ is paired with a compact convex set $\mathcal{X}_v$ and a point x_v contained in the set. Additionally, each vertex v is associated with a vertex cost $l_v(x_v)$, a nonnegative convex function of the point $x_v \in \mathcal{X}_v$. Finally, the edges of the graph are associated to convex constraints of the form $(x_u, x_v) \in \mathcal{X}_e$ which couple the vertices in an edge e.

A path p in the graph G is defined as a sequence of distinct vertices that connects a source vertex $s \in \mathcal{V}$ to the target vertex $t \in \mathcal{V}$. Let $\mathcal{P}$ denote the family of all s-t paths in the graph, and let $\mathcal{E}_p$ denote the set of edges traversed by the path p. The Shortest Path Problem (SPP) in GCS problem can then be stated as:

$$\begin{aligned}
\text{minimize} \quad & \sum_{v \in p} l_v(x_v) && \text{(1a)}\\
\text{subject to} \quad & p \in \mathcal{P}, && \text{(1b)}\\
& x_v \in \mathcal{X}_v, \quad \forall x_v \in p && \text{(1c)}\\
& (x_u, x_v) \in \mathcal{X}_e, \quad \forall e := (x_u, x_v) \in \mathcal{E}_p && \text{(1d)}
\end{aligned}$$

Problem (1) can be solved exactly as a Mixed-Integer Convex Program (MICP). Importantly, the problem formulation has a very tight convex relaxation, and can in many instances be solved to global optimality by solving the convex relaxation and performing a cheap rounding step on the integer variables, as shown in [52]. In this work, we treat the GCS framework as a modelling language that allows us to formulate and efficiently solve the problem presented in (1). Additionally, the GCS framework naturally gives us an upper bound on the optimality gap to a solution; Let $C_{\text{relax}} \leq C_{\text{opt}} \leq C_{\text{round}}$ be the costs of the relaxation, the MICP, and the rounded solution, respectively. The optimality gap δ_{opt} can then be overestimated as

$$\delta_{\text{opt}} = \frac{C_{\text{round}} - C_{\text{opt}}}{C_{\text{opt}}} \leq \frac{C_{\text{round}} - C_{\text{relax}}}{C_{\text{relax}}} = \delta_{\text{relax}} \qquad (2)$$

Finally, we note that the original problem description in [1], [52] has costs on edges rather than vertices, of which vertex costs is a special case.

B. Semidefinite relaxation of quadratically constrained quadratic programs

In this section we review nonconvex QCQPs and their semidefinite relaxation [53], and refer the reader to [54], [55], [56] for a more detailed discussion. This semidefinite relaxation and related techniques are commonly used in combinatorial optimization (see, e.g., [57]), and have been used in robotics for both localization and pose estimation [58], [59], [60], as well as motion planning around obstacles [61].

Let $y \in \mathbb{R}^n$ and let $x \coloneqq \begin{bmatrix} 1 & y^\intercal \end{bmatrix}^\intercal$. A nonconvex QCQP in homogeneous form is the optimization program

$$\begin{aligned} \text{minimize} \quad & x^\intercal Q_0 x && \text{(3a)} \\ \text{subject to} \quad & x^\intercal Q_i x \geq 0, \quad \forall i = 1, \ldots, l && \text{(3b)} \\ & Ax \geq 0 && \text{(3c)} \end{aligned}$$

where $A \in \mathbb{R}^{m \times (n+1)}$, and $Q_i \in \mathbb{R}^{(n+1)\times(n+1)}$, $i = 0, \ldots, l$. Note that positive-semidefiniteness is not required for Q_i, hence (3) can be nonconvex. A lower bound to (3) can be obtained by solving the semidefinite program [53], [55]:

$$\begin{aligned} \text{minimize} \quad & \operatorname{tr}(Q_0 X) && \text{(4a)} \\ \text{subject to} \quad & \operatorname{tr}(Q_i X) \geq 0, \quad \forall i = 1, \ldots, l && \text{(4b)} \\ & AXe_1 \geq 0 && \text{(4c)} \\ & AXA^\intercal \geq 0 && \text{(4d)} \\ & X = \begin{bmatrix} 1 & y^\intercal \\ y & Y \end{bmatrix} \succeq 0 && \text{(4e)} \end{aligned}$$

where e_1 is the $(n+1)$-vector with a 1 as the first component, and all the rest being zero.

Recalling that if the PSD matrix X is rank one, then $X = xx^T$, and using the simple identity $x^\intercal Q_i x = \operatorname{tr}(Q_i x x^\intercal)$, it can be seen that any feasible, rank-one solution to (4) provides a feasible solution to (3). Therefore (4) is indeed a relaxation. Moreover, if X^* is a rank-one minimizer of (4), then x^* satisfying $X^* = x^*(x^*)^\intercal$ is the global optimum of (3). This motivates the following scheme for obtaining near-optimal solutions to (3): After solving (4) to obtain an X^*, we take $\hat{x} = \begin{bmatrix} 1 & \hat{y}^\intercal \end{bmatrix}^\intercal$ to be the first column of X^*. We then use $\hat{x}$ as an initial guess for solving (3) using a local nonconvex solver. If a feasible solution is found, then we have the bound $C_{\text{relax}} \leq C_{\text{opt}} \leq C_{\text{round}}$ where C_{relax}, C_{opt}, and C_{round} are the optimal costs of the semidefinite relaxation, the nonconvex QCQP, and the rounded solution, respectively. While this strategy is not guaranteed to find a feasible solution to (3), our results in Section VIII demonstrate that in the case of planar pushing, this strategy finds both feasible and near-optimal solutions for all the tested problem instances.

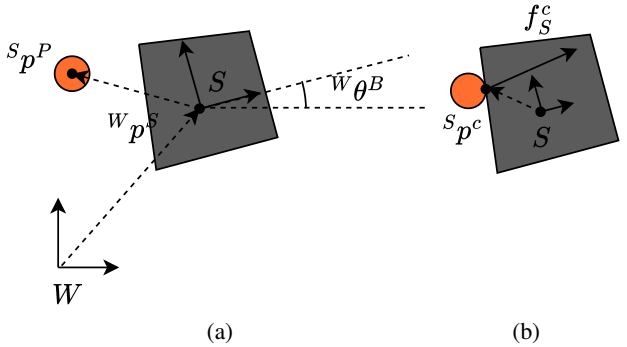
VI. The Slider-Pusher System

This section introduces the model for the slider-pusher system. We first describe the kinematics for the slider-pusher system, and then describe the model of the contact interaction between the slider and the underlying surface, known as the limit surface. Then, we present the full quasi-static dynamics model along with the frictional constraints.

We use the monogram notation from [62, §3.1] for reference frames. Denote the position of the frame B relative to A, measured in frame C by ${}^A p^B_C$ (when $C = A$, we write simply ${}^A p^B$). We similarly write ${}^A v^B_C$ (or ${}^A v^B$) for translational velocities, and ${}^A V^B_C$ (or ${}^A V^B$) for spatial velocities. The rotation that transforms a vector from frame B to frame A is denoted by ${}^A R^B$. To denote a Cartesian force applied at a point p measured in frame A we will write f^p_A, and to denote the spatial force we write F^p_A. We will drop the frames when they are clear from the context.

A. Kinematics

Consider the slider-pusher system in Figure 2 (a). Let W be the world frame, and attach the reference frames S and P at the COM of the slider and pusher, respectively. Let the pose of the slider in the world frame be given by $({}^W p^S, {}^W \theta^S) \in \mathrm{SE}(2)$. Let the (planar) spatial velocity of the slider be ${}^W V^S = ({}^W v^S, {}^W \omega^S) \in \mathrm{se}(2)$ (the tangent space associated with $\mathrm{SE}(2)$), where ${}^W v^S = (d/dt)\, {}^W p^S$ and ${}^W \omega^S = (d/dt)\, {}^W \theta^S$. Similarly, we represent the position of the i-th vertex of the slider with ${}^W p^{\nu_i}$, and its velocity by ${}^W v^{\nu_i}$, where $i = 1, \ldots, N_\nu$, and N_ν denotes the number of slider vertices. We represent the position of the pusher in the frame of the slider by ${}^S p^P \in \mathbb{R}^2$, and denote its velocity by ${}^S v^P = (d/dt)\, {}^S p^P$. We do not assign an orientation to the pusher, as it is assumed to be circular and hence invariant to rotation. When there is contact between the slider and the pusher, we will denote the position of the contact point in the slider frame by ${}^S p^c \in \mathbb{R}^2$ and the velocity by ${}^S v^c = (d/dt)\, {}^S p^c$ Going forward we will use these definitions, but will often drop the frames to keep notation light. We use (p^S, θ) and $V = (v^S, \omega^S)$ to denote the slider pose and spatial velocity in the world frame, and p^P, v^P, p^c, and v^c to denote the pusher position, translational pusher velocity, contact point position, and contact point velocity in the frame of the slider.



(a) (b)

Fig. 2: a) The slider-pusher kinematic quantities. b) The contact point and the contact forces.

B. The limit surface

The limit surface is a convex geometric object that describes the relationship between the applied spatial contact force on an object that is resting on a frictional surface and its instantaneous spatial velocity under the assumption of quasi-static dynamics [48]. In this work, we adapt the commonly used ellipsoidal approximation of the limit surface to model the interaction between the contact force applied by the pusher and the resulting spatial slider velocity [41], [63], [64].

Let $f^c_S \in \mathbb{R}^2$ be the contact force applied by the pusher measured in the slider frame, and let $\tau^c_S \in \mathbb{R}$ be the resulting

contact torque. Let $F_S^c \in \mathbb{R}^3$ denote the (planar) spatial contact force, given by:

$$F_S^c = \begin{bmatrix} f \\ \tau \end{bmatrix} = (J_S^c)^T f, \quad J_S^c = \begin{bmatrix} 1 & 0 & -p_y^c \\ 0 & 1 & p_x^c \end{bmatrix} \tag{5}$$

where $J_S^c \in \mathbb{R}^{2\times 3}$ denotes the contact Jacobian, and the frames have been dropped on the contact force and torque. From now, we will write simply f, τ, and F. See Figure 2 (b) for an illustration of these quantities.

We let the *limit surface* be defined by the ellipsoid in $\mathbb{R}^3$ described by $H(F) = (1/2)F^T DF = 1$, where

$$D = \text{diag}(c_f^{-1}, c_f^{-1}, c_\tau^{-1}) \in \mathbb{R}^{3\times 3} \tag{6}$$

$$c_f = \mu_S m_S g, \; c_\tau = cr\mu_S m_S g \tag{7}$$

and $\mu_S \in \mathbb{R}$ is the friction coefficient between the slider and table, $m_S \in \mathbb{R}$ is the slider mass, $g \in \mathbb{R}$ is the gravitional acceleration, $r \in \mathbb{R}$ is a characteristic distance, typically chosen as the max distance between a contact point and origin of frame S, and $c \in [0, 1]$ is an integration constant that depends on the slider geometry, as detailed in [63] and [65]. From the principle of maximum dissipation, it can be shown that when the slider is sliding, the applied spatial contact force F must lie on the limit surface, and the instantaneous spatial velocity V must be perpendicular to the limit surface: $V \propto \nabla H(F) = DF$, where $\nabla(\cdot)$ denotes the gradient operator and F and V are measured in the same frame [48]. We note that this can be approximated as the linear constraint $V = DF$ and appropriately scaling F as a post-processing step.

C. Slider-pusher dynamics

Define the state of the slider-pusher system as $x = (p^S, \theta, p^P)$ and the input as $u = (f, v^P)$. Let $\phi : \mathbb{R}^2 \to \mathbb{R}$ denote the Signed Distance Function (SDF) between the slider and pusher, a function of only the pusher position in the slider frame. Using the limit surface approximation from the previous subsection, the dynamics for the slider-pusher system can be written as:

$$\dot{x} = g(x, u) = \begin{bmatrix} RV_S \\ v^P \end{bmatrix} = \begin{bmatrix} RDF_S \\ v^P \end{bmatrix} \tag{8}$$

$$0 \le \phi(p^P) \perp \|f\|_2 \ge 0 \tag{9}$$

where

$$R = \begin{bmatrix} {}^W R^S & 0 \\ 0 & 1 \end{bmatrix}, \quad {}^W R^S \in \text{SO}(2)$$

transforms the spatial velocity from the slider frame into the world frame. Condition (9) is a complimentarity constraint that captures the fact that there can be no contact force when the pusher is not in contact with the slider.

D. Frictional constraints

The dynamics in (8) and (9) do not enforce that the contact force between the slider and pusher lies in the friction cone. Let $\lambda_n, \lambda_f \in \mathbb{R}$ be the normal and frictional components of f, that is, the components perpendicular and parallel to the contact face, respectively. Coloumb's friction law then states:

$$\lambda_n \ge 0, \quad |\lambda_f| \le \mu\lambda_n \tag{10}$$

where $\mu \in \mathbb{R}$ is the friction coefficient between the slider and pusher. Note that, subject to (10), the condition $\|f\|_2 = 0$ from the complimentarity constraint in (9) is equivalent to $\lambda_n = 0$.

Further, Coulomb's friction law states that when there is sliding motion between two objects, the friction force must be opposing the direction of motion, and lie on the boundary of the friction cone. Let ${}^S v^{c_\perp} \in \mathbb{R}$ be the tangential contact velocity in the slider frame. The contact modes are then given by:

- *Sticking.* When the contact between the pusher and slider is sticking, the relative tangential velocity must be zero

$$ {}^S v^{c_\perp} = 0 \tag{11}$$

- *Sliding Left.* When the contact point is sliding left relative to the slider, the friction force must lie on the boundary of the friction cone:

$$ {}^S v^{c_\perp} \le 0, \quad \lambda_f = -\mu\lambda_n \tag{12}$$

- *Sliding Right.* When the contact point is sliding right, the relationship is flipped:

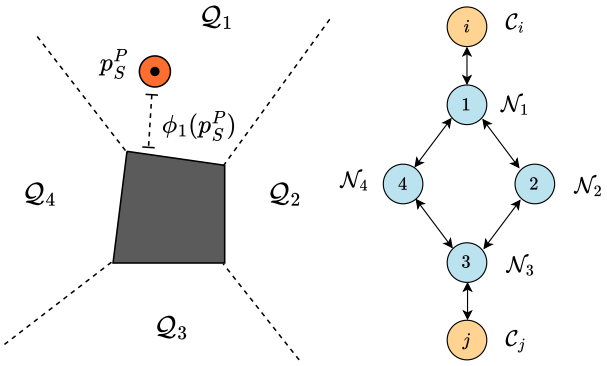
$$ {}^S v^{c_\perp} \ge 0, \quad \lambda_f = \mu\lambda_n \tag{13}$$

VII. Motion Planning for Planar Pushing

In this section, we present the motion planning algorithm. First, we decompose the configuration space of the slider-pusher system into a collection of subsets that we call *modes*. For each face of the slider, we have two types of modes: contact and non-contact. We then show how to formulate the motion planning problem within a mode as a convex program. Next, we formulate the motion planning problem as an SPP in a GCS, by constructing a graph where vertices correspond to modes of the system, edges correspond to valid mode transitions, and a path corresponds to a mode sequence and the motion for each mode in the sequence.

A. Modes of the slider-pusher system

The complimentarity constraint in (9) describes two classes of modes for the system: *non-contact* when $\phi(p^P) > 0$ and *contact* when $\phi(p^P) = 0$. The planar slider geometry is assumed a polygon, and we denote its N_F faces by $\mathcal{F}_1, \ldots, \mathcal{F}_{N_F}$. We can then partition the collision-free configuration space for the pusher $\mathcal{Q} = \{p^P \mid \phi(p^P) \ge 0\} \subseteq \mathbb{R}^2$ into N_F collision-free, possibly overlapping, polyhedral regions $\mathcal{Q}_1, \ldots, \mathcal{Q}_{N_F} \subseteq \mathcal{Q}$ in such a way that for each $\mathcal{Q}_i$, $i = 1, \ldots, N_F$ one face of the slider is represented by a supporting hyperplane to $\mathcal{Q}_i$. We additionally approximate ϕ by a piecewise linear function according to the decomposition $\mathcal{Q}_1, \ldots, \mathcal{Q}_{N_F}$ by associating a linear function ϕ_i to each $\mathcal{Q}_i$, $i = 1, \ldots, N_F$.



(a) (b)

Fig. 3: a) An example of a configuration-space partitioning $\mathcal{Q}_1, \ldots, \mathcal{Q}_4$ and the linear approximations $\phi_1, \ldots, \phi_4$ for a slider with convex planar geometry. b) The graph of non-contact modes that is added between every pair of vertices corresponding to contact modes $\mathcal{C}_i$ and $\mathcal{C}_j$. Corresponding modes are written in text next to the vertices.

In Figure 3 (a) we provide an example for a convex slider geometry, where one collision-free region is defined as the outside of each face such that two adjacent sets $\mathcal{Q}_i$ and $\mathcal{Q}_j$ are "split" by the vector $(\hat{n}_i + \hat{n}_j)/2$, where $\hat{n}_i$, $\hat{n}_j$ denotes the normal vectors of face $\mathcal{F}_i$ and $\mathcal{F}_j$. We can then define ϕ_i as the Euclidean distance from p^P to the half-plane that describes $\mathcal{F}_i$. When the closest point on the slider to p^P is on a face we have $\phi_i(p^P) = \phi(p^P)$, and $\phi_i(p^P) = 0$ only when the pusher is in contact with the slider. We note that in the case of a non-convex slider geometry a similar (but slightly more complex) decomposition of $\mathcal{Q}$ and approximation of ϕ is possible.

When the slider is not in contact with the pusher, the contact force must be zero due to (9), and (8) then implies that the slider pose is constant. For each of the collision-free configuration-space regions $\mathcal{Q}_i$ we therefore define a non-contact mode $\mathcal{N}_i$ by the constraint:

$$p^P \in \mathcal{Q}_i \tag{14a}$$

and the last row of eq. (8). Hence, for the non-contact modes we do not model the slider velocity or contact forces, and given a feasible initial slider pose that satisfies (16), the non-contact dynamics are described by linear constraints.

Similarly, for each face $\mathcal{F}_i$, $i = 1, \ldots, N_F$, on the planar slider geometry we define a contact mode $\mathcal{C}_i$ by the constraints (5), (8), (10), (11), and

$$\phi_i(p^P) = 0 \tag{15}$$

For simplicity, we do not consider sliding contact modes between the pusher and slider.

Finally, we represent θ^S with two variables $c_\theta, s_\theta \in \mathbb{R}$, with the additional constraint:

$$r \coloneqq (c_\theta, s_\theta), \quad \|r\|_2^2 = 1 \iff \begin{bmatrix} c_\theta & -s_\theta \\ s_\theta & c_\theta \end{bmatrix} \in \mathrm{SO}(2) \tag{16}$$

It now follows that the non-contact modes are described by linear constraints, and the contact modes are described by linear and (non-convex) quadratic constraints.

B. Motion planning within a mode

We discretize the dynamics in (8) using forward Euler discretization. We represent a trajectory segment within each mode for the slider-pusher system by N discrete knot points for the state and $N-1$ knot points for the input: $x_0, x_1, \ldots, x_N$ and $u_0, u_1, \ldots, u_{N-1}$. The dynamics are then $x_{k+1} = x_k + hg(x_k, u_k)$ for $k = 0, \ldots, N-1$ with the chosen time step $h \in \mathbb{R}_{>0}$.

Our formulation allows any cost that is a combination (or subset) of arc length and trajectory energy for the slider and pusher, contact forces, and time spent in (and near) contact. The cost takes the following form:

$$\sum_{k=0}^{N-1} L(x_k, x_{k+1}) + E(x_k, x_{k+1}, u_k) + k_f h \|f_k\|_2^2 + \sum_{k=0}^{N} \psi(x_k) \tag{17}$$

where the terms model the total trajectory arc length on SE(2) for the pusher and slider, the total kinetic energies for the pusher and slider trajectories, contact force regularization, and time spent near contact, respectively. Specifically

$$L(x_k, x_{k+1}) = k_{p^P} \|p_{k+1}^P - p_k^P\|_2 + k_{p^S}(1/N_\nu) \sum_{i=1}^{N_\nu} \|p_{k+1}^{\nu_i} - p_k^{\nu_i}\|_2 \tag{18}$$

models the total arc lengths on SE(2) of the pusher and slider, and

$$E(x_k, x_{k+1}, u_k) = k_{v^P} \|v_k^P\|_2^2 + k_{v^S}(1/N_\nu) \sum_{i=1}^{N_\nu} \|v_k^{\nu_i}\|_2^2 \tag{19}$$

models the total kinetic energies of the trajectories for the pusher and slider. Recall that p^{ν_i} and v^{ν_i} refer to the position and velocity for slider vertex $i = 1, \ldots, N_\nu$ in the world frame, where N_ν is the number of slider vertices. The total arc length of the slider trajectory on SE(2) is modelled through the mean arc length traversed by its vertices. Likewise, the energy of the slider trajectory is modelled through the mean squared velocity of its vertices. The constants k_{p^P}, k_{p^S}, k_{v^S}, k_{v^P} and $k_f \in \mathbb{R}_{>0}$ are cost weights for the arc lengths for the pusher and slider, trajectory energies for the pusher and slider, and contact force regularization.

The last term in (17) is a penalty on the distance between the pusher and the closest face of the slider. It is given by:

$$\psi(x) = \max_{i=1,\ldots,N_F} \frac{hk_T}{1 + \frac{1}{k_\phi}\phi_i(p^P)} \tag{20}$$

where k_T, $k_\phi \in \mathbb{R}_{>0}$ are cost weights for the time in contact and for being "close" to the slider, respectively. This term is at its maximum when there is contact between the pusher

and slider, and increases monotonically as the pusher gets closer to the slider. It has the property that $\psi(x) = hk_T$ when $\min_i \phi_i(x) = 0$, i.e. the cost of contact between the pusher and slider is k_T per second spent in contact. It also has the property that increasing k_ϕ increases the cost for being at a given distance to the object, i.e. k_ϕ controls the notion of "close". Since the denominator is always positive, this function is a maximum over convex functions and is thus readily encoded through N_F Rotated Second-Order Cone (RSOC) constraints [66].

We define the motion planning problem for a contact mode as minimizing (17) subject to (5), (8), (10), (11), (15), (16), and initial and final conditions on the state. This motion planning problem is non-convex due to the quadratic equality constraints arising from (5), (8) and (16). Specifically, the problem is a non-convex QCQP, which we relax into an SDP according to Subsection V-B. Further, we define the motion planning problem for a non-contact mode as minimizing (17) subject to the constraint (14a), the last row of (8), a constant slider pose, and initial and final conditions on the pusher position. Since these are all linear constraints, motion planning for a non-contact mode is naturally a convex program.

C. Constructing the graph of convex sets

Finally, we formulate the motion planning problem as an SPP in a GCS. Let $G = (\mathcal{V}, \mathcal{E})$ be the graph underlying a GCS as defined in Subsection V-A. For every contact mode $\mathcal{C}_i$, $i = 1, \ldots, N_F$ in the slider-pusher system, we add its corresponding SDP as a vertex v to $\mathcal{V}$, in the sense that we let $\mathcal{X}_v$ and l_v be the feasible set and cost function of the SDP, respectively. The point $x_v \in \mathcal{X}_v$ now corresponds to a trajectory of length N of states and inputs for the slider-pusher system in mode $\mathcal{C}_i$.

Transitions between contact modes require the pusher to move within the non-contact modes, from one face of the slider to another. Since the pusher can visit every non-contact mode in each of these trajectories, we add to the graph a copy of all the non-contact modes for every possible contact-mode transition. Specifically, for any two contact modes $\mathcal{C}_i$ and $\mathcal{C}_j$ where $i, j \in \{1, \ldots, N_F\}$, $i \neq j$, we create a graph $G_{ij} = (\mathcal{V}_{ij}, \mathcal{E}_{ij})$ where $\mathcal{V}_{ij}$ contains a vertex v for each non-contact mode $\mathcal{N}_k$, $k = 1, \ldots, N_F$ with $\mathcal{X}_v$ and l_v taken as the feasible set and cost function for the mode, and $\mathcal{E}_{ij}$ contains a bi-directional edge between all non-contact modes with intersecting collision-free regions. The vertices and edges of $\mathcal{G}_{ij}$ are added to G, as well as a bi-directional edge connecting the vertices corresponding to contact modes $\mathcal{C}_i$ and $\mathcal{C}_j$ to their respective non-contact modes to encode transition between contact and non-contact. An illustration is shown in Figure 3 (b). The total number of vertices in the graph is then $\mathcal{O}((N_F)^3)$, with N_F contact modes and $N_F\binom{N_F}{2}$ non-contact modes.

We enforce continuity between the state trajectories on a path in the graph. Specifically, for each edge $e = (u, v) \in \mathcal{E}$ we enforce that the last state in the trajectory in vertex u is equal to the first state in the trajectory in vertex v. Finally, we let the source vertex s and target vertex t be singleton sets that denote the desired starting and target states of the slider-pusher system, respectively. A feasible path p through G then has the interpretation as a continuous trajectory from the initial state to the target state, that consists of distinct trajectory segments for each mode represented by the vertices on the path, and the trajectory within each mode is determined by the points in the vertices.

D. Additional remarks

In the semidefinite relaxation of the QCQP, (4d) are new constraints formed by multiplying the linear constraints from the original problem together. This is a general recipe; we can potentially continue to multiply constraints together to obtain tighter relaxations at the cost of generating higher-degree polynomial constraints, using the Sums-of-Squares (SOS) / moment hierarchy [67]. Indeed, recent applications of the SOS hierarchy in motion planning have suggested that these higher-degrees were necessary to get tight relaxations [68]. In contrast, in this work we are able to obtain tight relaxations using only quadratic constraints, avoiding the significant computational cost of using higher-order relaxations. We accomplish this by writing additional tightening constraints implied by eq. (16).

First, we leverage (16) (which, for general rotation matrices looks like $R^\intercal R = I$) and add the dynamic constraints in (8) in both the world frame and the slider frame. While these constraints are obviously redundant for the QCQP due to (16), they yield different constraints in the semidefinite relaxation. Second, we use the initial and target rotations to generate tightening constraints. In (16) we parametrize the slider rotation by two variables in $\mathbb{R}^2$, where the start and target constraints on θ^S correspond to two points r_s and r_t on the unit circle. We define a hyperplane between these points in $\mathbb{R}^2$ with the normal vector pointing away from the origin, and add the constraint that r_k must be on the outside of this hyperplane for $k = 1, \ldots, N$. Under the assumption that an optimal plan should always pick the shortest geodesic path between two elements of $\mathrm{SO}(2)$, this constraint is redundant for the QCQP where (16) holds, but is not redundant in the relaxation.

Additionally, we exploit structure in our problem that allows us to write a computationally cheaper semidefinite relaxation in (4). Specifically, the motion planning problem for a contact mode exhibits a band-sparse structure, in the sense that the nonconvex quadratic constraints (5), (8), (16) only couple succeeding knot points x_k, x_{k+1}, and u_k, for $k = 0, \ldots, N-1$. This allows us to form the relaxation (4) as an SDP with several smaller PSD matrices instead of a single, larger matrix. In principle, this does not include all the tightening constraints (4d) and yields a potentially weaker convex relaxation, but in practice, we find that the loss in tightness is negligible while significantly reducing the computational cost. For more details, the reader is referred to e.g. [69].

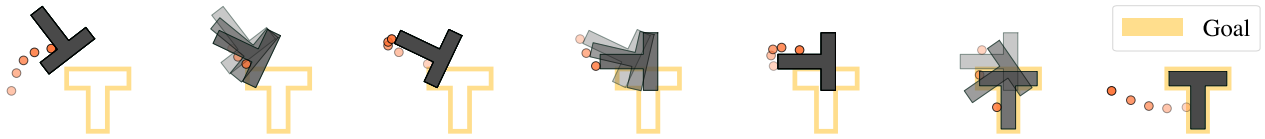


Fig. 4: Our planner simultaneously reasons about both discrete mode switches and continuous motion. Here, an example of a planar pushing plan with multiple mode switches is shown for a T-shaped slider geometry.

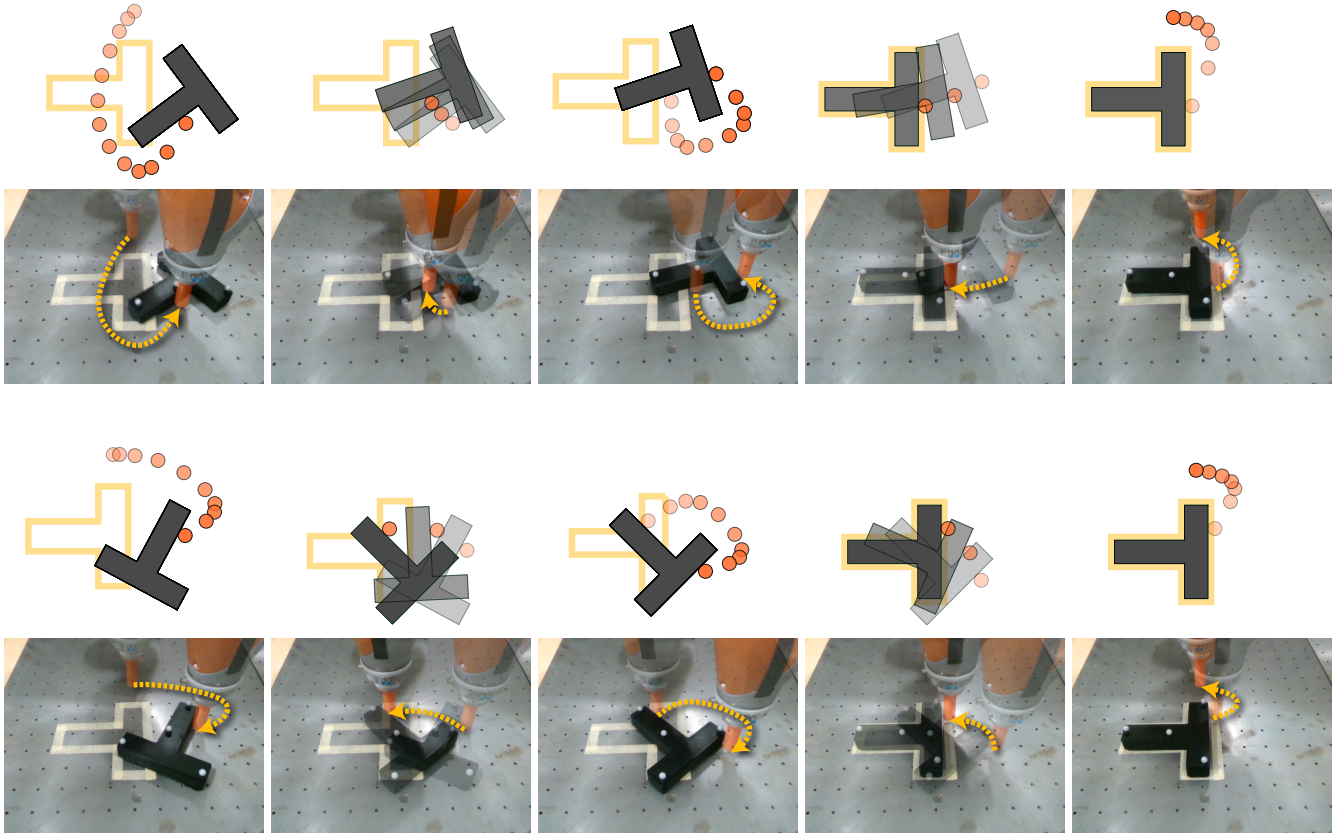


Fig. 5: Our method is able to generate close-to globally optimal plans for pushing tasks with collision-free motion planning between contact modes. Here, two different pushing trajectories for a T-shaped slider are shown, stabilized with a feedback controller on a real robotic system.

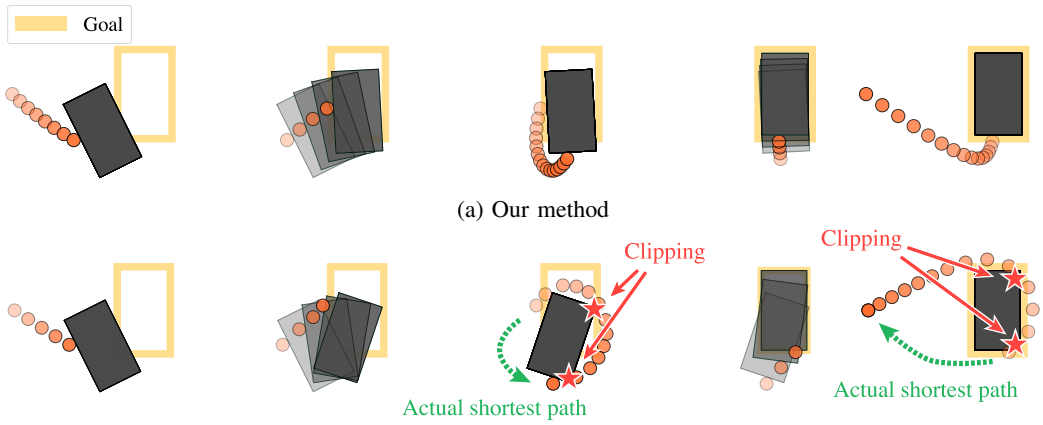
VIII. Experiments

This section contains both numerical and hardware experiments. An open-source implementation of the proposed motion planning algorithm, as well as the code used to run the experiments, can be found at `https://bernhardgraesdal.com/rss24-towards-tight-convex-relaxations/`.

We show an example of a generated motion plan in Figure 4. We choose the cost weights for contact modes to $k_{p^P} = k_{p^S} = k_{v^P} = 10$, $k_{v^S} = 100$, $k_f = 10$, $k_T = 1$, and $k_\phi = 0.1$ for the non-contact modes, where the only tuning that has been done is choosing reasonable orders of magnitude for the cost parameters, ensuring that the cost terms contribute approximately equally to the cost. We consider two different slider geometries, one simple box-shaped geometry and a T-shaped geometry, with the COM located according to a uniform mass distribution. For reference, both slider geometries have a maximum "radius" close to 0.25 meters, and the plans are generated within a box with sides 0.6 meters.

A. Planner performance

We evaluate the global optimality of the motion planner by generating 100 motion plans for the two slider geometries, with random initial and target configurations drawn from an uniform distribution, and upper-bound their respective optimality gaps according to (2). We use the conic solver Mosek



(a) Our method

(b) Contact-implicit method

Fig. 6: A comparative example between our method and a contact-implicit method. Our method picks the shortest path around the object, while the baseline goes the longer way around twice, highlighting the fact that our method is capable of global reasoning. Our method also guarantees that the trajectory stays collision-free between contacts, while the baseline can be seen to clip the corners of the slider.

[70] to solve the optimization problem on a laptop with an Apple M1 Max chip with 32GB of RAM. For both slider geometries, we achieve a success rate of 100%, that is, the rounding step is able to retrieve a feasible solution for all the generated problem instances. We show the average planning times, average rounding time, and average optimality gaps in Table I. We achieve a mean upper bound on the optimality gap of around 10%, with a median optimality gap of less than 8% for both slider geometries. We generate plans in a few seconds for the box-shaped slider, and in approximately 1.5 minutes for the T-shaped slider. The mean solve time for the box-shaped slider is an order of magnitude lower than for the T-shaped slider, which has twice as many faces, and hence generates a much bigger GCS problem.

Slider	SDP solve time	Rounding time	Optimality gap (δ_{round})
Box	7.05s (6.87s)	0.05s (0.05)s	8.33% (5.39%)
Tee	83.61s (80.12)	0.36s (0.014s) s	10.41% (7.47%)

TABLE I: Solve times, rounding times, and upper bounds on the global optimality gaps for planning planar pushing tasks with two different slider geometries, for 100 randomly chosen initial and final configurations. The box-shaped slider has 4 faces and the T-shaped slider has 8 faces. The reported values are mean values, with the median values shown in parenthesis.

Next, we show statistics for the generated convex optimization problems for the two different slider geometries in Table II. It is clear that the size of the optimization problem grows quickly (although polynomially) with the number of faces on the slider. We believe that we can significantly reduce the size of the optimization problem and thus reduce solve times by better exploiting the heavily structured problem formulation, but leave this for future work.

Problem statistics	Box	Tee
Number of constraints	48 846	212 795
Number of scalar variables	8 854	39 078
Number of PSD matrices	88	368

TABLE II: Statistics for the convex optimization problem that our methods formulates. The size of each PSD matrix is 15×15, and 3 knot points are used for each mode.

B. *Comparison with contact-implicit trajectory optimization*

To compare our method with a state-of-the-art baseline for contact-rich planning, we select a direct, contact-implicit trajectory optimization method similar to those proposed in [32] and [34]. This method encodes contact modes implicitly using complementarity constraints that are relaxed and solves the problem as a sequence of NLPs with increasingly strict complementarity constraints until they are satisfied with equality. This method requires an initial guess, which we provide as a straight-line interpolation between the initial and target configuration. We generate another 100 random initial and target configurations from an uniform distribution, and solve the planning problems with both methods.

	Success Rate	
Slider	Our method	Contact-implicit method
Box	100%	58%
Tee	100%	12%

TABLE III: A comparison of the success rate between our method and a contact-implicit trajectory optimization method. As our method is capable of global reasoning and does not rely on an initial guess, it has a much higher success rate compared to the baseline.

Table III presents a comparison between our method and the contact-implicit method. With a cost function that is specifically tuned to optimize the baseline's performance,

our method finds a solution in 100% of the instances for both slider geometries. In contrast, the baseline often fails, finding a solution in 58% of the instances for the box-shaped slider geometry and a mere 12% for the T-shaped slider. For the T-shaped slider, the contact-implicit method appears to succeed only when the initial and target configurations require a small number of mode switches that are relatively close to the initial guess. This limitation is not surprising, as the baseline is a local method that relies heavily on its initial guess. The T-shaped slider has a significantly higher number of possible mode sequences and a more complex SDF compared to the box-shaped geometry, making it more difficult for the baseline to find a solution. On the other hand, our method solves all instances for both geometries. This highlights a key advantage of our approach: by reasoning on a global level, our method (empirically) always finds a solution, without relying on an initial guess.

The two transcriptions rely on different pre-specified parameters, such as the number of knot points and trajectory/mode durations. These differences make it challenging to conduct a rigorous comparison of the plan quality between the two methods. Instead, we provide a qualitative analysis to highlight some advantages of our method. We start by pointing out that the performance of the contact-implicit method in table III required significant tuning of the cost function and problem numerics. After this tuning however, the found plans empirically seem very reasonable. In contrast, our method requires minimal tuning and no initial guess while still finding high-quality plans for all given problem instances. Our method also provides a good metric for evaluating solution quality through the obtained upper bound on the optimality gap, unlike the baseline, where such an objective comparison is difficult.

Furthermore, the baseline requires careful consideration to avoid problems with unwanted contacts and penetration between discretization points. While these problems can certainly be addressed by e.g. finer trajectory discretization, velocity constraints, or tuning, these issues are naturally handled in our method, which guarantees that the entire path is collision-free between contacts, independently of the aforementioned factors. Additionally, while the baseline requires a fixed, pre-specified trajectory duration, our method can vary the trajectory length through the number of visited contact modes. However, it is important to note that our method still relies on a pre-specified trajectory duration in each contact mode. Figure 6 illustrates some of the mentioned qualitative differences through a comparison of trajectories generated by the two methods.

C. Execution on real hardware

Finally, we demonstrate the feasibility of the obtained motion plans on a Kuka LBR iiwa 7 R800 7-DOF robotic arm, with a T-shaped slider object. A picture of the experimental setup is shown in Figure 1. We experimentally determine approximate values for the friction parameters from a handful of open-loop executions, and find the friction coefficient and integration constant for the contact between the slider and table in (7) to be $\mu_S = 0.5$ and $c = 0.3$, and the friction coefficient between the pusher and slider in (10) to be $\mu = 0.05$. We find that the plans generated with these friction parameters often perform well open-loop. For extra stability, we use a feedback controller to execute the plans on hardware, and employ a hybrid Model-Predictive Controller (MPC) commonly used for pushing tasks [42], [51]. Two different plans and the corresponding hardware execution can be seen in Figure 5, and a collection of hardware demonstrations can be seen in the video at `https://bernhardgraesdal.com/rss24-towards-tight-convex-relaxations/`.

IX. Conclusion and Future Work

In this work, we present a framework for planning near-globally optimal trajectories for contact-rich systems. We demonstrate how the dynamics of a contact-rich system can be described as a QCQP, and leverage semidefinite relaxations to approximate the planning problem in a fixed contact mode as a convex program. This convex approximation of the dynamics enables us to plan complex trajectories over mode sequences by leveraging the SPP in GCS framework from [1]. The entire planning problem can therefore be solved as a single convex optimization followed by a cheap rounding strategy based on local optimization. As an initial application, we study our framework on the planar pushing problem and demonstrate that our method empirically generates near-globally optimal solutions. We are currently working on applying our method to more complex contact tasks.

While the proposed approach is more efficient than prior global optimization methods, the planning times are still too slow for fast, reactive planning. We are currently working on better exploiting our highly structured problem formulation, both through the use of a custom solver, more compact problem formulations, and more efficient construction of the underlying optimization problem. Additionally, there is a rich literature on both exploiting special structure particularly in SDPs [69], [71], as well as approximating SDPs using simpler cones [72], [73] which could potentially reduce solve times by several orders of magnitude. Future work will explore the ability of these reduction methods to accelerate the planning. In addition, we are exploring how further techniques from the literature on SDP relaxations can be used to automatically provide even tighter relaxations in a reasonable computational budget.

Acknowledgement

This research was supported by (in alphabetical order): Aker Scholarship; Amazon.com Services LLC, PO No. 2D-12585006; Boston Dynamics AI Institute; and Office of Naval Research, Award No. N00014-22-1-2121. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] T. Marcucci, J. Umenberger, P. Parrilo, and R. Tedrake, "Shortest paths in graphs of convex sets," *SIAM Journal on Optimization*, vol. 34, no. 1, pp. 507–532, 2024.

[2] M. T. Mason, "Mechanics and Planning of Manipulator Pushing Operations," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 53–71, Sept. 1986.

[3] H. J. Suh, M. Simchowitz, K. Zhang, and R. Tedrake, "Do differentiable simulators give better policy gradients?" in *International Conference on Machine Learning*. PMLR, 2022, pp. 20 668–20 696.

[4] H. J. T. Suh, T. Pang, and R. Tedrake, "Bundled gradients through contact via randomized smoothing," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4000–4007, 2022.

[5] Q. L. Lidec, L. Montaut, C. Schmid, I. Laptev, and J. Carpentier, "Leveraging randomized smoothing for optimal control of nonsmooth dynamical systems," *arXiv preprint arXiv:2203.03986*, 2022.

[6] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[7] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[8] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," *arXiv preprint arXiv:1709.10087*, 2017.

[9] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3389–3396.

[10] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Proceedings of The 2nd Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Billard, A. Dragan, J. Peters, and J. Morimoto, Eds., vol. 87. PMLR, 29–31 Oct 2018, pp. 651–673. [Online]. Available: https://proceedings.mlr.press/v87/kalashnikov18a.html

[11] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[12] T. Chen, J. Xu, and P. Agrawal, "A system for general in-hand object re-orientation," in *Conference on Robot Learning*. PMLR, 2022, pp. 297–307.

[13] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik, "General in-hand object rotation with vision and touch," 2023.

[14] Z.-H. Yin, B. Huang, Y. Qin, Q. Chen, and X. Wang, "Rotating without seeing: Towards in-hand dexterity through touch," *arXiv preprint arXiv:2303.10880*, 2023.

[15] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," 2023.

[16] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, D. Sadigh, C. Finn, and S. Levine, "Octo: An open-source generalist robot policy," https://octo-models.github.io, 2023.

[17] N. M. M. Shafiullah, A. Rai, H. Etukuru, Y. Liu, I. Misra, S. Chintala, and L. Pinto, "On bringing robots home," *arXiv preprint arXiv:2311.16098*, 2023.

[18] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," *arXiv preprint arXiv:2205.09991*, 2022.

[19] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in *Conference on Robot Learning*. PMLR, 2022, pp. 158–168.

[20] X. Cheng, E. Huang, Y. Hou, and M. T. Mason, "Contact Mode Guided Sampling-Based Planning for Quasistatic Dexterous Manipulation in 2D," May 2021, pp. 6520–6526.

[21] ——, "Contact mode guided motion planning for quasidynamic dexterous manipulation in 3d," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE Press, 2022, p. 2730–2736. [Online]. Available: https://doi.org/10.1109/ICRA46639.2022.9811872

[22] W. Vega-Brown and N. Roy, "Asymptotically optimal planning under piecewise-analytic constraints," in *Algorithmic Foundations of Robotics XII: Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics*. Springer, 2020, pp. 528–543.

[23] C. Chen, P. Culbertson, M. Lepert, M. Schwager, and J. Bohg, "Trajectotree: Trajectory optimization meets tree search for planning multi-contact dexterous manipulation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8262–8268.

[24] N. Chavan-Dafle, R. Holladay, and A. Rodriguez, "Planar in-hand manipulation via motion cones," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 163–182, 2020. [Online]. Available: https://doi.org/10.1177/0278364919880257

[25] A. Wu, S. Sadraddini, and R. Tedrake, "R3t: Rapidly-exploring random reachable set tree for optimal kinodynamic planning of nonlinear hybrid systems," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4245–4251.

[26] T. Pang, H. J. T. Suh, L. Yang, and R. Tedrake, "Global Planning for Contact-Rich Manipulation via Local Smoothing of Quasi-dynamic Contact Models," June 2022.

[27] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Transactions on Graphics (ToG)*, vol. 31, no. 4, pp. 1–8, 2012.

[28] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 4906–4913.

[29] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1168–1175.

[30] I. Chatzinikolaidis and Z. Li, "Trajectory optimization of contact-rich motions using implicit differential dynamic programming," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2626–2633, 2021.

[31] T. A. Howell, S. Le Cleac'h, S. Singh, P. Florence, Z. Manchester, and V. Sindhwani, "Trajectory optimization with optimization-based dynamics," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6750–6757, 2022.

[32] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, Jan. 2014.

[33] J.-P. Sleiman, J. Carius, R. Grandia, M. Wermelinger, and M. Hutter, "Contact-implicit trajectory optimization for dynamic object manipulation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6814–6821.

[34] Z. Manchester and S. Kuindersma, "Variational contact-implicit trajectory optimization," in *Robotics Research: The 18th International Symposium ISRR*. Springer, 2020, pp. 985–1000.

[35] W. Jin and M. Posa, "Task-Driven Hybrid Model Reduction for Dexterous Manipulation," Nov. 2022.

[36] M. Wang, A. Ö. Önol, P. Long, and T. Padır, "Contact-Implicit Planning and Control for Non-prehensile Manipulation Using State-Triggered Constraints," in *Robotics Research*, ser. Springer Proceedings in Advanced Robotics, A. Billard, T. Asfour, and O. Khatib, Eds. Cham: Springer Nature Switzerland, 2023, pp. 189–204.

[37] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. Del Prete, "Optimization-based control for dynamic legged robots," *IEEE Transactions on Robotics*, 2023.

[38] T. Marcucci and R. Tedrake, "Mixed-integer formulations for optimal control of piecewise-affine systems," in *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, 2019, pp. 230–239.

[39] T. Marcucci, R. Deits, M. Gabiccini, A. Bicchi, and R. Tedrake, "Approximate hybrid model predictive control for multi-contact push recovery in complex environments," in *2017 IEEE-RAS 17th international conference on humanoid robotics (Humanoids)*. IEEE, 2017, pp. 31–38.

[40] B. Aceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernández-López, and C. Semini, "Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2531–2538, 2017.

[41] F. R. Hogan and A. Rodriguez, "Reactive planar non-prehensile manipulation with hybrid model predictive control," *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 755–773, June 2020.

[42] ——, "Feedback Control of the Pusher-Slider System: A Story of Hybrid and Underactuated Contact Dynamics," in *Algorithmic Foundations of Robotics XII*, K. Goldberg, P. Abbeel, K. Bekris, and L. Miller, Eds., vol. 13. Cham: Springer International Publishing, 2020, pp. 800–815.

[43] Z. Liu, G. Zhou, J. He, T. Marcucci, L. Fei-Fei, J. Wu, and Y. Li, "Model-based control with sparse neural dynamics," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[44] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *2014 IEEE-RAS international conference on humanoid robots*. IEEE, 2014, pp. 279–286.

[45] B. Aceituno-Cabezas and A. Rodriguez, "A Global Quasi-Dynamic Model for Contact-Trajectory Optimization in Manipulation," in *Robotics: Science and Systems XVI*. Robotics: Science and Systems Foundation, July 2020.

[46] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, "A convex model of humanoid momentum dynamics for multi-contact motion generation," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, Nov. 2016, pp. 842–849.

[47] F. A. Koolen, "Balance control and locomotion planning for humanoid robots using nonlinear centroidal models," Thesis, Massachusetts Institute of Technology, 2020.

[48] S. Goyal, A. Ruina, and J. Papadopoulos, "Planar sliding with dry friction Part 1. Limit surface and moment function," *Wear*, vol. 143, no. 2, pp. 307–330, Mar. 1991.

[49] K. M. Lynch and M. T. Mason, "Stable Pushing: Mechanics, Controllability, and Planning," *The International Journal of Robotics Research*, vol. 15, no. 6, pp. 533–556, Dec. 1996.

[50] N. Doshi, F. R. Hogan, and A. Rodriguez, "Hybrid Differential Dynamic Programming for Planar Manipulation Primitives," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 6759–6765.

[51] F. R. Hogan, E. R. Grau, and A. Rodriguez, "Reactive Planar Manipulation with Convex Hybrid MPC," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 247–253.

[52] T. Marcucci, M. Petersen, D. von Wrangel, and Russ Tedrake, "Motion planning around obstacles with convex optimization," *Science Robotics*, vol. 8, no. 84, p. eadf7843, 2023.

[53] N. Z. Shor, "Quadratic optimization problems," *Soviet Journal of Computer and Systems Sciences*, vol. 25, pp. 1–11, 1987.

[54] J. Park and S. P. Boyd, "General Heuristics for Nonconvex Quadratically Constrained Quadratic Programming," *arXiv: Optimization and Control*, Mar. 2017.

[55] P. A. Parrilo, "Lecture notes from MIT 6.7230 - Algebraic techniques and semidefinite optimization," 2023.

[56] H. D. Sherali and W. P. Adams, *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*, ser. Nonconvex Optimization and Its Applications, P. Pardalos and R. Horst, Eds. Boston, MA: Springer US, 1999, vol. 31.

[57] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM (JACM)*, vol. 42, no. 6, pp. 1115–1145, 1995.

[58] H. Yang, J. Shi, and L. Carlone, "TEASER: Fast and Certifiable Point Cloud Registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, Apr. 2021.

[59] F. Dümbgen, C. Holmes, and T. D. Barfoot, "Safe and Smooth: Certified Continuous-Time Range-Only Localization," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1117–1124, Feb. 2023.

[60] M. P. Giamou, "Semidefinite Relaxations for Geometric Problems in Robotics," Thesis, Mar. 2023.

[61] B. El Khadir, J. B. Lasserre, and V. Sindhwani, "Piecewise-linear motion planning amidst static, moving, or morphing obstacles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7802–7808.

[62] R. Tedrake, *Robotic Manipulation*, 2023. [Online]. Available: http://manipulation.mit.edu

[63] K. 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*, vol. 1. Raleigh, NC: IEEE, 1992, pp. 416–421.

[64] N. Chavan-Dafle, R. Holladay, and A. Rodriguez, "Planar in-hand manipulation via motion cones," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 163–182, Mar. 2020.

[65] N. Chavan Dafle, R. Holladay, and A. Rodriguez, "In-Hand Manipulation via Motion Cones," *Robotics: Science and Systems XIV*, June 2018.

[66] S. Boyd and L. Vandenberghe, *Convex Optimization*.

[67] J. B. Lasserre, "The moment-sos hierarchy," in *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*. World Scientific, 2018, pp. 3773–3794.

[68] S. Teng, A. Jasour, R. Vasudevan, and M. Ghaffari, "Convex geometric motion planning on lie groups via moment relaxation," *arXiv preprint arXiv:2305.13565*, 2023.

[69] M. Garstka, M. Cannon, and P. Goulart, "A clique graph based merging strategy for decomposable SDPs," May 2020.

[70] M. ApS, *MOSEK Fusion API for C++ 10.1.24*, 2024. [Online]. Available: https://docs.mosek.com/latest/cxxfusion/index.html

[71] S. Burer and R. D. Monteiro, "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," *Mathematical programming*, vol. 95, no. 2, pp. 329–357, 2003.

[72] A. Majumdar, G. Hall, and A. A. Ahmadi, "Recent scalability improvements for semidefinite programming with applications in machine learning, control, and robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 331–360, 2020.

[73] G. Blekherman, S. S. Dey, M. Molinaro, and S. Sun, "Sparse psd approximation of the psd cone," *Mathematical Programming*, vol. 191, no. 2, pp. 981–1004, 2022.