Dexterous Contact-Rich Manipulation via the Contact Trust Region

H.J. Terry Suh*,[1], Tao Pang*,[2], Tong Zhao[2] and Russ Tedrake[1]

***Abstract*—What is a good local description of contact dynamics for contact-rich manipulation, and where can we trust this local description? While many approaches often rely on the Taylor approximation of dynamics with an ellipsoidal trust region, we argue that such approaches are fundamentally inconsistent with the unilateral nature of contact. As a remedy, we present the Contact Trust Region (CTR), which captures the unilateral nature of contact while remaining efficient for computation. With CTR, we first develop a Model-Predictive Control (MPC) algorithm capable of synthesizing local contact-rich plans. Then, we extend this capability to plan globally by stitching together local MPC plans, enabling efficient and dexterous contact-rich manipulation. To verify the performance of our method, we perform comprehensive evaluations, both in high-fidelity simulation and on hardware, on two contact-rich systems: a planar IiwaBimanual system and a 3D AllegroHand system. On both systems, our method offers a significantly lower-compute alternative to existing RL-based approaches to contact-rich manipulation. In particular, our Allegro in-hand manipulation policy, in the form of a roadmap, takes fewer than 10 minutes to build offline on a standard laptop, with online inference taking just a few seconds. Experiment data, video and code are available at ctr.theaiinstitute.com.**

I. Introduction

ROBOTS today rarely leverage their embodiment to the fullest due to the limitations of our computational algorithms: robot arms only establish contact with the end-effectors and only perform collision-free motion planning, and robot hands often only establish contact with the fingertips instead of leveraging the entire surface of the hand. This stands in stark contrast to humans, as we are able to utilize every part of our body to strategically establish contact with the environment. In order to address this gap, dexterous contact-rich manipulation, where a robot must autonomously decide where to establish contact without restricting possible contacts, remains an important problem for us to solve.

At the heart of many iterative algorithms for manipulation lies the question: what is a good local description of contact mechanics for contact-rich manipulation that i) faithfully captures local behavior, and ii) is sufficiently simple for efficient and scalable computation? Classically, many works have utilized Jacobian-based analysis to locally reason about how the contact forces affect kinematic changes in configurations [1]–[3]. These mechanics have been used heavily as local representations of contact for planning and control in subsequent works. For instance, classical grasping analysis and synthesis relies on local wrench-space arguments [4], [5]. Many direct methods for gradient-based trajectory optimization encode Jacobian-based local dynamics as constraints as well [6]–[8].

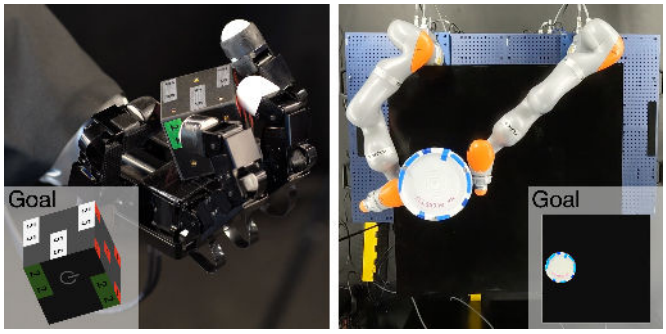


Fig. 1: Hardware experiments illustrating the utility of our proposed method in contact-rich manipulation. Left: Dexterous in-hand manipulation with the Allegro hand moving a cube. Right: Whole-body manipulation with bimanual iiwas moving a bucket.

At surface level, these approaches seem different from those that use a differentiable simulator [9]–[15] to build Taylor approximations of the dynamics, thus abstracting away the details of contact mechanics from the planning algorithms. Previous works have utilized this locally-linear first-order Taylor approximation for the purposes of trajectory optimization and control [11], [16]–[20].

The apparent discrepancy between the two approaches begs the question of how they are related. In fact, deriving the gradients given by the differentiable simulators analytically reveals that the locally linear model and the local Jacobian-based models are fundamentally related [11, Example 5]. However, the form of each approximation hides a more fundamental difference when we ask: *where can we trust this local model*? For the local approximations, this region is known as the *trust region* in optimization literature [21]. Intuitively, the trust region describes where the local model closely approximates the original function, letting us safely rely on it for local improvements. As the quality of Taylor approximations typically degrades as we move further from the nominal point, previous works have often utilized an *ellipsoidal trust region* [21]–[23].

Our first contribution (Section III) is to elucidate the inconsistency between the ellipsoidal trust region and the *unilateral* nature of contact. We then remedy this inconsistency by delving deeper into the structure of modern differentiable simulators and proposing the *contact trust region*. The contact trust region allows us to connect representations based on differentiable simulators to those in classical works, such as the wrench set or the motion set, and can be readily obtained by combining ingredients from existing differentiable simulators.

Our analysis is based on quasidynamic differentiable simu-

*Equal contributions. [1]Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology. [2]Boston Dynamics AI Institute. Correspondence to hjsuh94@csail.mit.edu.

This work is funded by .

lator proposed in [11], called the Convex Quasidynamic Differentiable Contact (CQDC) contact model. The CQDC model, recapped in Section II, is very representative of approaches taken by modern differentiable simulators through contact [11], [12], [24], [25]. It treats simulation of contact as a convex optimization problem whose primal solution becomes the next state x_+, and gives gradients by performing sensitivity analysis in convex optimization.

However, while a bulk of previous approaches [11], [17]–[19] have only focused on building a Taylor approximation of the configurations based on sensitivity gradients, our *contact trust region* also builds a linear model over the contact impulses by utilizing sensitivity analysis to obtain gradients with respect to dual variables. Having a linear model over the contact impulses allow us to further qualify the ellipsoidal trust region by imposing friction cone constraints which are often present in works that directly encode contact mechanics.

The contact trust region and its connection to convex formulations of contact dynamics also allows us to seamlessly integrate the advances in contact smoothing for planning and control [7], [11], [12], [18], [26]. Without contact smoothing, many classical Jacobian-based methods as well as local trajectory optimizers are limited to fixed contact mode sequences. Thus, their ability to describe contact mechanics locally has been limited to describing the relationship between possible contact forces and changes in state within a contact mode. The myopicness of this local model is consistent with the *non-smooth* nature of contact, which also spells trouble for gradient-based Taylor approximations of contact dynamics.

While complementarity-based representations [27] and mixed-integer formulations [28]–[30] attempt to alleviate this limitation by considering contact mode changes globally, optimizing through these models often require combinatorial search through all possible contact mode sequences. Given the near-exponential scalability of contact modes [31], these methods are therefore not fit to tackle problems that are rich in contact. Thus, many successful algorithms go through an approximate representation in which complementarity is omitted in practice [32]–[34].

To alleviate these limitations, smoothing-based methods relax contact dynamics by reasoning about the average result of all the contact modes around a nominal state given some distribution of state and action disturbances. Recent works [9], [18] have attributed the success and scalability of Reinforcement Learning (RL) to this mechanism. Surprisingly, however, [11], [35] shows that for contact dynamics simulated by convex optimization, this process can be done without Monte-Carlo by using *barrier smoothing* which performs interior-point relaxation of the original contact dynamics and produces force from a distance. Yet, naively applying classical control methods such as LQR on smoothed linearization of contact dynamics has shown mixed results [36].

Our proposed contact trust region advances the discussion on smoothing by indicating where the smoothed dynamics remain reliable. As our approach draws directly on convex optimization concepts, particularly the sensitivity analysis of primal and dual solutions, we can enjoy the same smoothing benefits by applying an interior-point relaxation that perturbs the original problem's KKT conditions.

As our second contribution, we present a highly-efficient contact-implicit Model Predictive Control (MPC) method [32], [37] in Section IV. Specialized to contact-rich manipulation, our MPC method leverages the contact trust region to faithfully approximate contact dynamics locally. By formulating the contact trust region as a convex set—specifically, an intersection of multiple second-order cone constraints—we can readily incorporate it into various local trajectory optimization problems such as goal tracking and force control [38], [39].

We show the efficacy of our proposed contact trust region and MPC method on two representative contact-rich manipulation problems in Figure 1: i) whole-body manipulation on the bimanual iiwa and bucket system, and ii) in-hand reorientation on the Allegro hand and cube system. For both systems, we aim to answer the following two questions: i) is the MPC method successful in planning for goals under quasidynamic equations of motion (Section V), and ii) can the MPC method successfully stabilize quasidynamic plans under true second-order dynamics (Section VI)? Through 2000 trajectory runs in simulation and 100 runs on hardware, our results delve into the efficacy and the limitations of our approach, as well as the benefits of the contact trust region over the ellipsoidal one in the context of planning and control.

While our MPC is highly effective in local planning, the goals it can achieve are fundamentally limited by local reachability. For further goals that require non-trivial exploration, its capacity is limited in the absence of a good initial guess. Furthermore, when the MPC encounters such goals, its performance is relatively hard to characterize.

As our final contribution, we address global planning by chaining local trajectories discovered by MPC. To do this, we follow a roadmap [40] approach where we seed each node with a stable object configuration. Then, we connect these nodes by first finding good actuator configurations for local MPC to start from using our contact trust region (Section VII), then rolling out the MPC plan. To demonstrate the constructed roadmap's robustness, we perform on the Allegro hand system 150 consecutive edge traversals before a hardware failure occurs. For new goals that are not in the roadmap, we simply find the closest node to the goal, then connect them by the same process of finding a good initial actuator configuration and then rolling out MPC.

Our proposed global planning method (Section VIII) enables efficient dexterous re-orientation on the Allegro hand, and has several advantages compared to existing approaches. Compared to single-query algorithms [11], [41]–[43], most of the computation time for our algorithm happens offline in the roadmap-building stage, enabling fast online inference. Moreover, the offline roadmap construction only takes minutes on a standard laptop, which is orders-of-magnitude less than approaches based on deep RL [44]–[46].

Through our three contributions of i) the contact trust region that approximates contact mechanics efficiently, ii) a highly effective gradient-based MPC controller specialized to local contact-rich manipulation, and iii) a global planner that stitches together local trajectories, our method succeeds in dramatically reducing computation time compared to state-of-the-art methods in RL [45]–[47] or sampling-based MPC methods [48], [49] without the need for parallelization. We

hope digging deep into the structure of contact mechanics can bring efficiency to the current state-of-the-art approaches for whole-body manipulation and dexterous in-hand reorientation.

II. Contact Dynamics as Convex Optimization

Many state-of-art methods for simulating contact solve a constrained optimization problem at every step [10]–[12], [50]–[53]. In this paper, we use the quasi-dynamic formulation of contact dynamics presented in [11], where we assume the system is highly damped by frictional forces.

A. Setup & Notation

We assume the state consists of configurations, which we denote by $q \in \mathbb{R}^{n_q}$, and omit velocities due to the quasi-dynamic assumption. These configurations are divided into actuated configurations $q^{\mathrm{a}} \in \mathbb{R}^{n_{q_\mathrm{a}}}$ that belong to the robot, and unactuated configurations $q^{\mathrm{u}} \in \mathbb{R}^{n_{q_\mathrm{u}}}$ which represent object dofs. Furthermore, the actions are represented as a position command $u \in \mathbb{R}^{n_{q_\mathrm{a}}}$ to a stiffness controller with a diagonal gain matrix $\mathbf{K}_\mathrm{a} \in \mathbb{R}^{n_{q_\mathrm{a}} \times n_{q_\mathrm{a}}}$.

B. Equations of Motion

At a configuration q, the equations of motion are framed as a search for next configurations $q_+ = (q_+^{\mathrm{a}}, q_+^{\mathrm{u}})$ such that the following balances of impulse holds,

$$h\mathbf{K}_\mathrm{a}(q_+^{\mathrm{a}} - u) = h\tau^{\mathrm{a}} + \sum_{i=1}^{n_c} (\mathbf{J}_{\mathrm{a}_i}(q))^\top \lambda_i \tag{1a}$$

$$\epsilon \mathbf{M}_\mathrm{u}(q) \frac{q_+^{\mathrm{u}} - q^{\mathrm{u}}}{h} = h\tau^{\mathrm{u}} + \sum_{i=1}^{n_c} (\mathbf{J}_{\mathrm{u}_i}(q))^\top \lambda_i \tag{1b}$$

where $h \in \mathbb{R}_{>0}$ is the timestep, $\tau^{\mathrm{a}} \in \mathbb{R}^{n_{q_\mathrm{a}}}$ and $\tau^{\mathrm{u}} \in \mathbb{R}^{n_{q_\mathrm{u}}}$ are external torques acting on actuated and unactuated bodies respectively (e.g. gravity), and $\epsilon \in \mathbb{R}_{\geq 0}$ is a regularization constant.

In addition, we define an index set $\mathcal{I}_c$ over pairs of contact geometries, which we can obtain from a collision detection algorithm. For each contact pair $i \in \mathcal{I}_c$, $\lambda_i \in \mathbb{R}^3$ correspond to contact impulses. By convention, we denote $\lambda_i = (\lambda_{\mathrm{n}_i}, \lambda_{\mathrm{t}_i})$ where $\lambda_{\mathrm{n}_i} \in \mathbb{R}$ corresponds to contact impulse along the normal direction of the contact frame, and $\lambda_{\mathrm{t}_i} \in \mathbb{R}^2$ corresponds to the frictional forces along the tangential plane.

We also define the contact Jacobians $\mathbf{J}_i \in \mathbb{R}^{3 \times (n_{q_\mathrm{u}} + n_{q_\mathrm{a}})}$ as local mappings from the configurations q to the contact frame. The top row corresponds to the normal direction of the contact frame, where as the bottom two rows orthogonally span the tangential plane. The Jacobians can further be decomposed into mappings into actuated and unactuated dofs,

$$\mathbf{J}_i \coloneqq [\mathbf{J}_{\mathrm{u}_i}, \mathbf{J}_{\mathrm{a}_i}] \coloneqq \begin{bmatrix} \mathbf{J}_{\mathrm{n}_i} \\ \mathbf{J}_{\mathrm{t}_i} \end{bmatrix} \in \mathbb{R}^{3 \times (n_{q_\mathrm{u}} + n_{q_\mathrm{a}})}. \tag{2}$$

With this notational setup, (1a) describes that the configuration of the actuated bodies at the next step, q_+^{a}, will be decided such that the sum of the impulses must be balanced by the impulse experienced by the stiffness controller at the next step. Similarly, (1b) states that the unbalanced impulses result in relative displacements of the unactuated bodies.

C. Contact Constraints

Additional constraints further qualify the relationship between the next configurations q_+ and contact impulses λ_i:

1) *non-penetration*: next configuration is non-penetrating,
2) *friction cone*: λ_i must be inside the friction cone,
3) *complementarity*: contact cannot be applied from a distance, and the direction of friction opposes the movement (i.e. maximum dissipation).

To impose these constraints, we first define the friction cone $\mathcal{K}_i^\star$ for each contact point, as well as its dual cone $\mathcal{K}_i$,

$$\mathcal{K}_i^\star \coloneqq \left\{ \lambda_i = (\lambda_{\mathrm{n}_i}, \lambda_{\mathrm{t}_i}) \in \mathbb{R}^3 | \mu_i \lambda_{\mathrm{n}_i} \geq \sqrt{\lambda_{\mathrm{t}_i}^\top \lambda_{\mathrm{t}_i}} \right\}, \tag{3a}$$

$$\mathcal{K}_i \coloneqq \left\{ \nu_i = (\nu_{\mathrm{n}_i}, \nu_{\mathrm{t}_i}) \in \mathbb{R}^3 | \nu_{\mathrm{n}_i} \geq \mu_i \sqrt{\nu_{\mathrm{t}_i}^\top \nu_{\mathrm{t}_i}} \right\}. \tag{3b}$$

Anistescu's relaxation allows us to write down these contact constraints in a convex manner by introducing a mild non-physical artifact, where sliding in the tangential plane results in separation in the normal direction,

$$\nu_i \coloneqq \mathbf{J}_i(q_+ - q) + [\phi_i, 0, 0]^\top \in \mathcal{K}_i, \tag{4a}$$

$$\lambda_i \in \mathcal{K}_i^\star, \tag{4b}$$

$$\nu_i^\intercal \lambda_i = 0. \tag{4c}$$

where ϕ_i is the current signed distance for contact pair i.

D. Contact Dynamics as Convex Optimization

We can frame the equations of motion with contact as a search for the next configurations q_+ that satisfy the constraints introduced in Section II-B and Section II-C,

$$\text{find} \quad q_+ \tag{5a}$$

$$\text{s.t.} \quad (1a), (1b), (4a), (4b), (4c). \tag{5b}$$

Remarkably, this is equivalent to the KKT conditions of the following Second-Order Cone Program (SOCP),

$$\min_{q_+} \frac{1}{2} {q_+}^\top \mathbf{P} q_+ + b^\top q_+, \text{ subject to} \tag{6a}$$

$$\mathbf{J}_i q_+ + c_i \in \mathcal{K}_i, \ \forall i \in \mathcal{I}_c, \text{ where} \tag{6b}$$

$$\mathbf{P} \coloneqq \begin{bmatrix} \epsilon \mathbf{M}_\mathrm{u}/h & 0 \\ 0 & h\mathbf{K}_\mathrm{a} \end{bmatrix}, \ b \coloneqq -h \begin{bmatrix} \epsilon \mathbf{M}_\mathrm{u} q^{\mathrm{u}}/h^2 + \tau^{\mathrm{u}} \\ \mathbf{K}_\mathrm{a} u + \tau^{\mathrm{a}} \end{bmatrix}, \tag{6c}$$

$$c_i \coloneqq [\phi_i, 0, 0]^\top - \mathbf{J}_i q. \tag{6d}$$

where $\mathcal{I}_c$ is the index set over potential contact pairs. Note that stationarity (1), primal feasibility (4a), dual feasibility (4b), and complementary slackness eq. (4c) are the optimality conditions of this SOCP, known as the Karush-Kuhn-Tucker (KKT) conditions. For complementary slackness of SOCPs, we note that we do not impose element-wise slackness: the primal and dual vectors can both be non-zero as long as they are orthogonal [11], [54].

In dynamical systems and control theory, it is conventional to represent the dynamic system as a map f that takes a state and input then maps it to the next state. As our state only

consists of configurations q for quasidynamic systems, we use the following notation for contact dynamics,

$$\begin{aligned} q_+ =& f(q,u) && \text{(7a)}\\ =& \operatorname{argmin}_{q_+} \ \frac{1}{2} q_+^\top \mathbf{P} q_+ + b^\top q_+, \\ & \text{subject to } \ \mathbf{J}_i q_+ + c_i \in \mathcal{K}_i, \forall i \in \mathcal{I}_c. && \text{(7b)} \end{aligned}$$

E. Sensitivity Analysis

Changing q, u will change the problem parameters of the optimization program (6). For instance, changing u will change b, and changing q will change $\mathbf{Q}, b, \mathbf{J}_i$, and c_i. Thus, the dynamics map describes a *solution map* from the change in parameters of the SOCP to the optimal solutions.

Various first-order methods in numerical optimization and optimal control often utilize the gradients of the dynamics map f with respect to the state q (velocities omitted due to the quasidynamic assumption) and input u. As our dynamics can be interpreted as a solution map, this question can be reduced to asking: how does the optimal solution of an optimization problem change as we change the problem parameters?

We can answer this question through the technique of *sensitivity analysis*, which states that as we locally perturb the problem parameters, the primal and dual solutions of (6) change in a way that preserves (up to first-order) the equality constraints involving stationarity (1) and complementarity (4c) constraints. This is equivalent to stating that the derivative of the equality constraints with respect to q or u are also zero at optimality. For instance, rewriting the derivative of (1) and (4c) in matrix form gives us

$$\begin{bmatrix} \mathbf{P} & \mathcal{H}_i \left[-\mathbf{J}_i\right]^\top \\ \mathcal{V}_i \left[\lambda_i^\top \mathbf{J}_i\right] & \mathcal{D}_i \left[\mathbf{J}_i q_+ + c_i\right] \end{bmatrix} \begin{bmatrix} \partial q_+ / \partial u \\ \mathcal{V}_i \left[\partial \lambda_i / \partial u\right] \end{bmatrix} = \begin{bmatrix} -\partial b / \partial u \\ 0 \end{bmatrix} \tag{8}$$

where $\mathcal{H}_i, \mathcal{V}_i, \mathcal{D}_i$ stands for horizontal, vertical, and diagonal stacking of the terms according to indices of *active* constraints (see Section IX-A for derivation). Then, we can use the implicit function theorem to solve the system of equations and obtain the derivatives $\partial q_+ / \partial u$ and $\partial \lambda_i / \partial u$. The derivatives with respect to q may be obtained in a similar fashion, but are more involved due to the $\partial \mathbf{J}_i / \partial q$ term (curvature).

F. Smoothing of Contact Dynamics

In sensitivity analysis, we fixed the active and inactive constraints: this is natural since there is no way to make an active constraint inactive and vice-versa *locally*. This gives us more insight into the non-smooth nature of contact, where the dynamics is continuous, piecewise smooth but has discontinuous gradients. Each smooth piece, also known as a *contact mode*, corresponds to a combination of active and inactive constraints [55].

Unfortunately, such non-smooth dynamics can be subject to fast-changing gradients that can destabilize methods that rely on local approximations [18]. In addition, the landscape can have flat regions if none of the contact modes are active, leaving optimizers stranded without informative directions of improvement [9]. As such, various dynamic smoothing methods have been proposed over the years to relax this numerical difficulty [7], [11], [12], [18], [26]

In [11], [12], a method of smoothing out optimization-based dynamics was introduced based on the log-barrier (interior-point) relaxation of (6), which solves for the KKT equations where the complementarity condition is perturbed by a positive constant κ,

$$\lambda_i^\top \left(\mathbf{J}_i q_+ + c_i\right) = \kappa. \tag{9}$$

The resulting solution is equivalent to solving the log-barrer relaxation of (6),

$$\min_{q_+} \ \frac{1}{2} q_+^\top \mathbf{P} q_+ + b^\top q_+ - \kappa \sum_{i=1}^{n_c} \log\left[\nu_{i,n}^2 - \mu \|\nu_{i,t}\|^2\right], \tag{10}$$

where $\nu_i = \mathbf{J}_i q_+ + c_i$. The resulting dynamics creates a force-field effect where bodies that are not in contact apply forces inversely proportional to their distance (9). The gradients of this smoothed dynamics can be obtained similarly to the SOCP case by differentiating the perturbed complementarity constraints (9) instead of the strict one (4c). However, unlike the SOCP case, every constraint is active in this relaxation, which intuitively agrees with the continuity of the gradients.

Similar to (7), we denote the dynamics map over the relaxed contact dynamics throughout the manuscript as

$$\begin{aligned} q_+ &= f_\kappa(q,u) && \text{(11a)}\\ &= \operatorname{argmin}_{q_+} \Big(\frac{1}{2} q_+^\top \mathbf{P} q_+ + b^\top q_+ \\ &\qquad - \kappa \sum_{i=1}^{n_c} \log\left[\nu_{i,n}^2 - \|\nu_{i,t}\|^2\right] \Big) && \text{(11b)} \end{aligned}$$

As an unconstrained optimization problem, (10) technically does not have dual variables. However, if we consider solving the SOCP dynamics (6) with the interior point method, (10) corresponds to one major iteration along the *central path* [54, §11.6]. Therefore, we can identify contact impulses with the *dual feasible points* along the central path:

$$\lambda_{\kappa,i} = \frac{\kappa}{\nu_{i,0}^2 - \mu \|\nu_{i,t}\|^2} \begin{bmatrix} \nu_{i,n} \\ -\mu \nu_{i,t} \end{bmatrix}. \tag{12}$$

We note that this impulse i) has a direction opposing the movement in the tangential plane, ii) has a magnitude that satisfies (9), and iii) lies in the friction cone.

III. Local Approximation of Contact Dynamics

Consider a simple, single-horizon optimization problem that involves the contact dynamics (6) as a constraint,

$$\begin{aligned} \min_{q_+, u} \quad & \|q_{\text{goal}} - q_+\|^2 && \text{(13a)}\\ \text{s.t.} \quad & q_+ = f(q,u). && \text{(13b)} \end{aligned}$$

A majority of first-order algorithms for optimization and optimal control rely on iteratively making a *local* approximation of the problem that is more amenable for computation. For instance, gradient-descent makes a linear approximation, and Newton's method makes a quadratic one. In this section, we answer the question: what is the right local approximation of the contact dynamics that i) is computationally convenient, and ii) captures the correct local behavior?

A. Linear Model over Smoothed Primal and Dual Variables

Previous works on smooth dynamical systems have often resorted to a first-order Taylor approximation of the dynamics around some nominal coordinates $(\bar{q}, \bar{u})$, and approximates the dynamics with a linear model. This Taylor approximation can be compactly written as

$$q_+ = \mathbf{A}\delta q + \mathbf{B}\delta u + f(\bar{q}, \bar{u}), \tag{14a}$$
$$\mathbf{A} \coloneqq \partial f/\partial q(\bar{q}, \bar{u}), \quad \mathbf{B} \coloneqq \partial f/\partial u(\bar{q}, \bar{u}), \tag{14b}$$
$$\delta q \coloneqq q - \bar{q}, \quad \delta u \coloneqq u - \bar{u}. \tag{14c}$$

However, many previous works [7], [9], [11], [12], [56] have noted that due to the non-smooth nature of contact, it is beneficial to smooth the dynamics where a first-order Taylor approximation would be valid beyond the contact mode to which $(\bar{q}, \bar{u})$ belongs [11]. A first-order Taylor expansion under such a smoothed dynamics model can be written as

$$q_+ = \mathbf{A}_\kappa \delta q + \mathbf{B}_\kappa \delta u + f_\kappa(\bar{q}, \bar{u}), \tag{15a}$$
$$\mathbf{A}_\kappa \coloneqq \partial f_\kappa/\partial q(\bar{q}, \bar{u}), \quad \mathbf{B}_\kappa \coloneqq \partial f_\kappa/\partial u(\bar{q}, \bar{u}), \tag{15b}$$
$$\delta q \coloneqq q - \bar{q}, \quad \delta u \coloneqq u - \bar{u}. \tag{15c}$$

where κ is the barrier smoothing parameter introduced in (9).

Still, as this linear model only gives a local approximation, we would naturally expect the quality of this approximation to degrade as we get further from the nominal coordinates $(\bar{q}, \bar{u})$. In order to more rigorously characterize the validity of the local linear model, we need to analyze the behavior of $(q_+, \lambda_{+,i})$ as we vary $(\delta q, \delta u)$, where $\lambda_{+,i}$ are the linearized dual variables defined by:

$$\lambda_{+,i} = \mathbf{C}_{\kappa,i}\delta q + \mathbf{D}_{\kappa,i}\delta u + \lambda_{\kappa,i}(\bar{q}, \bar{u}), \tag{16a}$$
$$\mathbf{C}_{\kappa,i} \coloneqq \partial \lambda_{\kappa,i}/\partial q(\bar{q}, \bar{u}), \quad \mathbf{D}_{\kappa,i} \coloneqq \partial \lambda_{\kappa,i}/\partial u(\bar{q}, \bar{u}). \tag{16b}$$

Lemma 1 (Taylor Approximation)**.** *Consider the joint linear model of the primal and dual variables,*

$$\begin{bmatrix} q_+ \\ \mathcal{V}_i[\lambda_{+,i}] \end{bmatrix} = \begin{bmatrix} \mathbf{A}_\kappa & \mathbf{B}_\kappa \\ \mathcal{V}_i[\mathbf{C}_{\kappa,i}] & \mathcal{V}_i[\mathbf{D}_{\kappa,i}] \end{bmatrix} \begin{bmatrix} \delta q \\ \delta u \end{bmatrix} + \begin{bmatrix} f_\kappa(\bar{q}, \bar{u}) \\ \mathcal{V}_i[\lambda_{\kappa,i}(\bar{q}, \bar{u})] \end{bmatrix}. \tag{17}$$

Then, this linear model satisfies the equality conditions of the perturbed SOCP (1b),(1a),(9) to first order,

$$\begin{bmatrix} \bar{\mathbf{P}} q_+ + \bar{b} - \sum_{i=1}^{n_c} \bar{\mathbf{J}}_i^\top \lambda_{+,i} \\ \mathcal{V}_i\left[(\bar{\mathbf{J}}_i q_+ + \bar{c}_i)^\top \lambda_{+,i} - \kappa\right] \end{bmatrix} = \mathcal{O}\left((\delta q, \delta u)^2\right) \tag{18}$$

with $\bar{\mathbf{P}} \coloneqq \mathbf{P}(\bar{q}, \bar{u}) + \frac{\partial \mathbf{P}}{\partial q}\delta q + \frac{\partial \mathbf{P}}{\partial u}\delta u$*, and vice-versa for* $\bar{b}, \bar{\mathbf{J}}, \bar{c}_i$.

Proof: If we expand the Taylor-approximated equations and drop the terms above first-order, we are left with i) the original equality conditions over nominal coordinates which must hold due to the nominal values being obtained at optimality and ii) its first-order expansion, which must hold due to the definition of the sensitivity gradients (8). ■

B. Ellipsoidal Trust Region (ETR)

Lemma 1 implies that the linear model will no longer be accurate far away from $(\bar{q}, \bar{u})$, even with the beneifts of smoothing [11]. As such, many optimization methods are further concerned with the question of where we can *trust* the local model, known as the *trust region* around $(\bar{q}, \bar{u})$ that further qualify the accuracy of the linear model [21], [23]. To simplify computation, these trust regions are often chosen to be simple geometric primitives.

Following classical works, let us first consider an ellipsoidal trust region which can be described in quadratic form:

$$\mathcal{E}_{\mathbf{\Sigma}}(\bar{q}, \bar{u}) \coloneqq \{\delta z = (\delta q, \delta u) | \delta z^\top \mathbf{\Sigma} \delta z \leq 1\}. \tag{19}$$

The set of q_+ achievable with this ellipsoidal trust region can be written as

$$\{q_+ | q_+ = \mathbf{A}_\kappa \delta q + \mathbf{B}_\kappa \delta u + f_\kappa(\bar{q}, \bar{u}), (\delta q, \delta u) \in \mathcal{E}_{\mathbf{\Sigma}}(\bar{q}, \bar{u})\} \tag{20}$$

In order for this linear model to be consistent with the true dynamics within the trust region, choosing an appropriate trust region parameter $\mathbf{\Sigma}$ is imperative. If $\mathbf{\Sigma}$ has too large of a volume, it is likely that there will be large errors in the behavior of the system within the trust region. On the other hand, if $\mathbf{\Sigma}$ is too small, iterative algorithms will have to take more iterations to converge.

C. The Feasible Trust Region (FTR)

However, contact is more special in that it is a *unilaterally constrained* dynamical system[1]. If points in the trust region predicts a behavior of primal and dual variables that are not feasible, this can also cause a large discrepancy in the approximated model within the trust region. In practice solvers would also try to find an ellipsoidal trust region such that all the elements within the trust region would result in a feasible prediction [57], [58]. However, this often also results in overly conservative and small region as we get closer to the boundary.

Thus, we propose to separate the role of the ellipsoid in keeping Taylor approximation error low, and having to be inscribed within the feasible set. This is achievable by intersecting the ellipsoidal trust region and the primal and dual feasible set under linearized approximations of the primal and dual variables. We formalize this object in Definition 1.

Definition 1 (Feasible Trust Region)**.** We define the **Feasible Trust Region** (FTR) at $(\bar{q}, \bar{u})$ as the set of all allowable perturbations that do not result in violation of the primal and dual feasibility constraints under a linear model,

$$\mathcal{F}_{\mathbf{\Sigma},\kappa}(\bar{q}, \bar{u}) \coloneqq \{(\delta q, \delta u) | \delta z^\top \mathbf{\Sigma} \delta z \leq 1, \delta z = (\delta q, \delta u), \tag{21a}$$
$$q_+ = \mathbf{A}_\kappa \delta q + \mathbf{B}_\kappa \delta u + f_\kappa(\bar{q}, \bar{u}), \tag{21b}$$
$$\lambda_{+,i} = \mathbf{C}_{\kappa,i}\delta q + \mathbf{D}_{\kappa,i}\delta u + \lambda_{\kappa,i}(\bar{q}, \bar{u}), \tag{21c}$$
$$\mathbf{J}_i q_+ + c_i \in \mathcal{K}_i, \tag{21d}$$
$$\lambda_{+,i} \in \mathcal{K}_i^\star\}. \tag{21e}$$

We note the FTR (21) is convex as an intersection of convex constraints. As such, it can be efficiently embedded into convex conic solvers as constraints. When iteratively solving (13), the FTR (21) allows us to take larger step sizes compared to the traditional ellipsoid trust region (19), as primal and dual feasibility are guaranteed up to first order with the addition of constraints (21d) and (21e). Larger step sizes beneficially reduce the required number of iterations, which

[1] bilateral constraints are often included in sensitivity analysis, so would be implicitly obeyed by the linear model

is ideal especially when fast and approximate solutions are desired.

We emphasize that the primal (21d) and dual (21e) feasibility constraints both need to be considered: some $(\delta q, \delta u)$ may end up in violation of primal feasibility by predicting penetration, yet might be dual-feasible as the required impulse still lies within the friction cone. Other perturbations may be primal feasible as it predicts a non-penetrating configuration, but may be dual infeasible by predicting pulling-like actions.

D. The Contact Trust Region (CTR)

While FTR (21) improves upon the classical ellipsoidal trust region by allowing larger steps, it can still face slower convergence if the feasible set itself is too small. Unfortunately, this is often the case if we impose all of the constraints in (21). To more accurately represent the object forward reachability, we leverage the fact that the configuration of the object is of much more importance compared to the robot, and selectively apply a subset of the constraints in (21).

The feasibility constraints in (21) can be grouped into primal and dual feasibility constraints. In addition, as each constraint is specific to a pair of collision geometries, the collision geometries can be grouped into geometries that belong to the robot (actuated geometries), the object (unactuated geometries) and the environment. This implies there are 6 possible combinations of these geometry pairs. We list our choice of which constraints are imposed and dropped in Table I.

	(u,u)	(a,a)	(e,e)	(u,a)	(u,e)	(a,e)
Primal Feasibility	O	X	X	X	O	X
Dual Feasibility	O	X	X	O	O	X

TABLE I: Choice of imposed (O) and dropped (X) constraints for each pair of groups of collision geometries.

In particular, we noticed that imposing the primal feasibility constraints between the robot and objects lead to too conservative of an approximation. This is due to the fact that under a linear model of the smoothed dynamics, the sensitivity of the actuated bodies is often much larger than the sensitivity of the unactuated one (note that this is a relaxation of the sensitivity of the unactuated body being $\mathbf{0}$ and the actuated body being $\mathbf{I}$ when not in contact) - thus, the actuated body catches up to penetrate the unactuated body upon very small perturbations. As the positions of where the robots end up at the next step is relatively less important than the object, we decide to drop this constraint. In contrast, primal feasibility constraints between the object and the environment should be encoded, as non-penetration with the environment qualifies the true set of allowable motion for the object. Likewise, we impose all dual constraints that directly interact with the unactuated body and ignore the rest of the dual constraints.

Definition 2 (Contact Trust Region)**.** Let $\mathcal{I}_{\text{uu}}$ be the index set of collision pairs between (unactuated, unactuated), $\mathcal{I}_{\text{ue}}$ for between (unactuated, environment), and $\mathcal{I}_{\text{ua}}$ for between (unactuated,actuated). Then, we define the **Contact Trust Region** (CTR) as

$$\mathcal{S}_{\mathbf{\Sigma},\kappa}(\bar{q},\bar{u}) \coloneqq \{(\delta q,\delta u) | \delta z^\top \mathbf{\Sigma} \delta z \le 1, \delta z = (\delta q, \delta u), \quad (22a)$$
$$q_+ = \mathbf{A}_\kappa \delta q + \mathbf{B}_\kappa \delta u + f_\kappa(\bar{q},\bar{u}), \quad (22b)$$
$$\lambda_{+,i} = \mathbf{C}_{\kappa,i}\delta q + \mathbf{D}_{\kappa,i}\delta u + \lambda_{\kappa,i}(\bar{q},\bar{u}), \quad (22c)$$
$$\mathbf{J}_i q_+ + c_i \in \mathcal{K}_i, \ \ \forall i \in \mathcal{I}_{\text{uu}} \cup \mathcal{I}_{\text{ue}}, \quad (22d)$$
$$\lambda_{+,i} \in \mathcal{K}_i^\star \ \ \forall i \in \mathcal{I}_{\text{uu}} \cup \mathcal{I}_{\text{ua}} \cup \mathcal{I}_{\text{ue}}\}. \quad (22e)$$

Furthermore, we make the following definition for the predicted dynamics over the trust region,

Definition 3 (Motion Set)**.** We define the image of $\mathcal{S}(\bar{q},\bar{u})$ under the linearized primal solution map as the **Motion Set**,

$$\mathcal{M}_{\mathbf{\Sigma},\kappa}(\bar{q},\bar{u}) \coloneqq \{q_+ | q_+ = \mathbf{A}_\kappa \delta q + \mathbf{B}_\kappa \delta u + f_\kappa(\bar{q},\bar{u}), \quad (23a)$$
$$(\delta q, \delta u) \in \mathcal{S}_{\mathbf{\Sigma},\kappa}(\bar{q},\bar{u})\}. \quad (23b)$$

Furthermore, we denote the projection of this set in the object subspace as the object motion set $\mathcal{M}^{\text{u}}_{\mathbf{\Sigma},\kappa}(\bar{q},\bar{u}) \coloneqq \{q^{\text{u}}_+ | q_+ \in \mathcal{M}_{\mathbf{\Sigma},\kappa}(\bar{q},\bar{u})\}$, and the actuated part as the robot motion set. Note that as a linear map of a convex set, the motion set (23) is also convex.

E. Mechanics Derivation of the Motion Set

Our choice of constraints in CTR can be further motivated by connecting the CTR to classical sets in manipulation, such as the motion cone [59], [60], or the wrench set [61], [62]. As these classic works are often concerned with geometric Jacobian-based analysis at the current configuration, we first introduce a special case of the CTR that has zero perturbations in configuration $(\delta q = 0)$.

Definition 4 (Action-only Contact Trust Region)**.** We define the Action-only Contact Trust Region (ACTR) $\mathcal{S}_{\mathbf{\Sigma}U,\kappa}$ as

$$\mathcal{S}_{\mathbf{\Sigma}U,\kappa}(\bar{q},\bar{u}) \coloneqq \{\delta u | \delta u^\top \mathbf{\Sigma} \delta u \le 1 \quad (24a)$$
$$q_+ = \mathbf{B}_\kappa \delta u + f_\kappa(\bar{q},\bar{u}) \quad (24b)$$
$$\lambda_{+,i} = \mathbf{D}_{\kappa,i}\delta u + \lambda_{\kappa,i}(\bar{q},\bar{u}) \quad (24c)$$
$$\mathbf{J}_i q_+ + c_i \in \mathcal{K}_i, \ \ \forall i \in \mathcal{I}_{\text{uu}} \cup \mathcal{I}_{\text{ue}} \quad (24d)$$
$$\lambda_{+,i} \in \mathcal{K}_i^\star \ \ \forall i \in \mathcal{I}_{\text{uu}} \cup \mathcal{I}_{\text{ua}} \cup \mathcal{I}_{\text{ue}}\}. \quad (24e)$$

We now show how we can derive the ACTR from classical sets in manipulation.

The Contact Impulse Set. For a single contact pair i, the predicted linear model of how the contact impulse changes as we vary δu is given by $\lambda_{+,i} = \mathbf{D}_{\kappa,i}\delta u + \lambda_{\kappa,i}(\bar{q},\bar{u})$, as long as the prediction of $\lambda_{+,i}$ lies within the friction cone $\mathcal{K}_i^\star$. Thus, the set of allowable contact impulses for this pair is given by

$$\mathcal{C}_{\mathbf{\Sigma}U,\kappa,i}(\bar{q},\bar{u}) \coloneqq \big\{\lambda_{+,i} | \lambda_{+,i} = \mathbf{D}_{\kappa,i}\delta u + \lambda_{\kappa,i}(\bar{q},\bar{u}) \quad (25a)$$
$$\lambda_{+,i} \in \mathcal{K}_i^\star \ \ \forall i \in \mathcal{I}_{\text{uu}} \cup \mathcal{I}_{\text{ua}} \cup \mathcal{I}_{\text{ue}} \quad (25b)$$
$$\delta u^\top \mathbf{\Sigma} \delta u \le 1\big\} \quad (25c)$$

The Generalized Friction Cone. The generalized friction cone [63] is a representation of the friction cone applied in the space of object coordinates. This can be obtained by applying the contact Jacobian that linearizes the kinematics of the contact point with respect to object coordinates,

$$\mathcal{JC}_{\mathbf{\Sigma}U,\kappa,i}(\bar{q},\bar{u}) \coloneqq \big\{w_i | w_i = \mathbf{J}_{\text{u}_i}^\top \lambda_{+,i}, \ \lambda_{+,i} \in \mathcal{C}_{\mathbf{\Sigma}U,\kappa,i}\big\} \quad (26a)$$

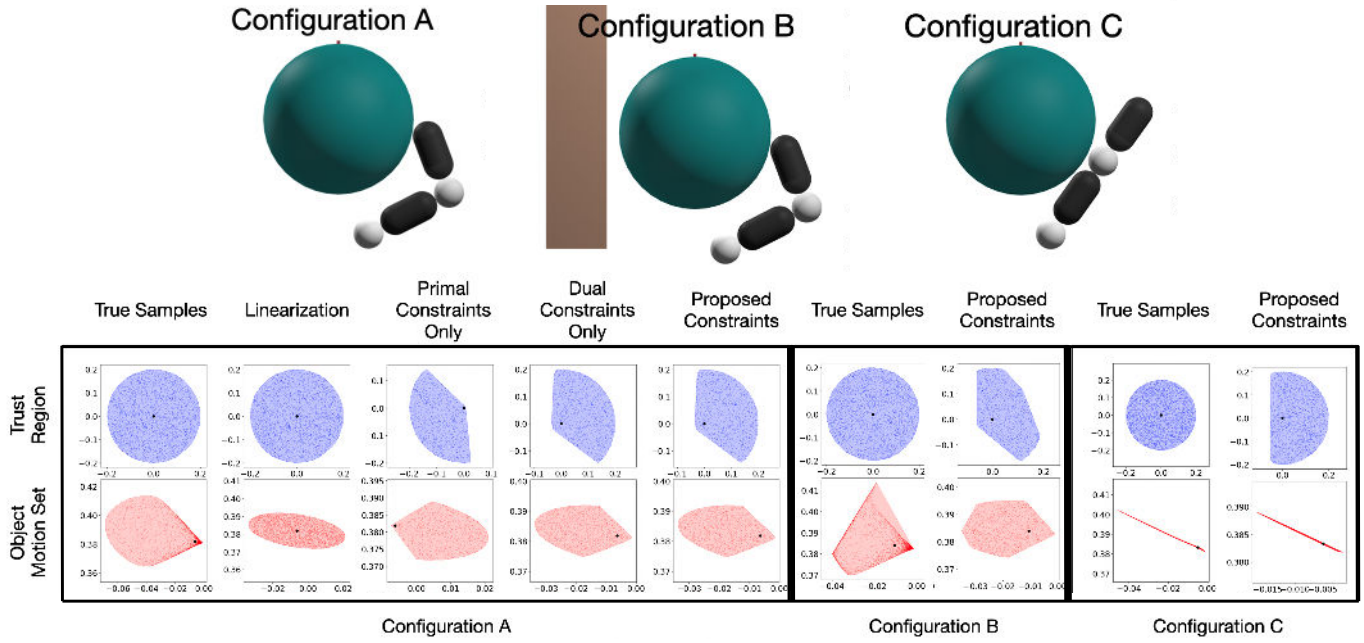


Fig. 2: Visualization of the action-only contact trust region $\mathcal{S}_{\mathbf{\Sigma}U,\kappa}$ and the object motion set $\mathcal{M}^{\mathrm{u}}_{\Sigma U}$ under different conditions.

The Wrench Set. The wrench set [61], [62] is defined as the set of all achievable wrenches that can be applied to an object given all possible contact forces that can be applied from a certain configuration. This quantity has classically served as an important metric in grasping analysis [5], [61], [64]. While the classical wrench set has bounds on the contact force, we will adopt this definition and apply bounds on the actuator input instead ($\|\delta u\|^2_{\mathbf{\Sigma}} \leq 1$).

Our version of the wrench set is defined by taking Minkowski sums of each generalized friction cone $\mathcal{JC}_{\mathbf{\Sigma}U,i}$ for all contact pairs,

$$\mathcal{W}_{\mathbf{\Sigma}U,\kappa}(\bar{q},\bar{u}) \coloneqq \{w | w = h\tau^{\mathrm{u}} + \textstyle\sum_i w_i \tag{27a}$$
$$w_i \in \mathcal{JC}_{\mathbf{\Sigma}U,\kappa,i}(\bar{q},\bar{u})\}, \tag{27b}$$

where $h\tau^{\mathrm{u}}$ is some external impulse on the object (e.g. by gravity). The wrench set adequately describes the set of all wrenches that can be applied on the object from a given nominal configuration and input $(\bar{q},\bar{u})$.

The Motion Set. The motion set is analogous to motion cones [59], [60] and is defined as the set of all feasible motions that can be achieved. We construct this object by utilizing a quasistatic relation between the object movement and the applied impulse (6b), and further imposing non-penetration constraints with the environment as well as other possible objects,

$$\mathcal{M}^{\mathrm{u}}_{\mathbf{\Sigma}U,\kappa}(\bar{q},\bar{u}) = \{q^{\mathrm{u}}_+ | \epsilon \mathbf{M}_{\mathrm{u}}(\bar{q}) \left(q^{\mathrm{u}}_+ - \bar{q}^{\mathrm{u}}\right)/h = w, \tag{28a}$$
$$w \in \mathcal{W}_{\mathbf{\Sigma}U,\kappa}(\bar{q},\bar{u}), \tag{28b}$$
$$\mathbf{J}_{\mathrm{u}_i}(q^{\mathrm{u}}_+ - \bar{q}^{\mathrm{u}}) + c_i \in \mathcal{K}_i, \tag{28c}$$
$$\forall i \in \mathcal{I}_{\mathrm{uu}} \cup \mathcal{I}_{\mathrm{ue}}\}. \tag{28d}$$

We can now draw the connection between the construction of motion sets in this section, and the definition of the motion set as the image of the trust region in Definition 3.

Lemma 2. The object motion set derived in (28) is equivalent to the Action-only Motion Set where Definition 3 is applied to (24),

$$\mathcal{M}^{\mathrm{u}}_{\Sigma U}(\bar{q},\bar{u}) = \{q_+ | q_+ = \mathbf{B}^{\mathrm{u}}_{\kappa}\delta u + f^{\mathrm{u}}_{\kappa}(\bar{q},\bar{u}) \tag{29a}$$
$$\delta u \in \mathcal{S}_{\mathbf{\Sigma}U,\kappa}(\bar{q},\bar{u})\} \tag{29b}$$

Proof: Section IX-B ■

F. Additional Constraints

We list some additional constraints and costs that may be added to this formulation to further restrict the CTR.

1) Joint Limits: Consider joint limits q^{a}_{lb} and q^{a}_{lb} that must be enforced by the robot. Then, we can add the constraint that limits the position command that we send to the robot,

$$q^{\mathrm{a}}_{lb} \leq \bar{u} + \delta u \leq q^{\mathrm{a}}_{ub}. \tag{30}$$

2) Torque Limits: Many manipulators such as the iiwas also have limits on the sensed torque experienced by the robots, which can be represented as lower and upper bounds τ_{lb} and τ_{ub}. Although the quasi-dynamic formulation cannot account for dynamic transient torques, we can compute the steady-state torque experienced by the robot using the difference in the sent position command and the predicted position [65]. Thus, the set of achievable δu that does not violate this steady-state torque limit can be written as

$$\tau_{lb} \leq \mathbf{K}^{-1}_{\mathrm{a}}\left(\bar{q}^{\mathrm{a}}_+ + \mathbf{B}^{\mathrm{a}}\delta u - \bar{u}\right) \leq \tau_{ub}. \tag{31}$$

where τ^{a} accounts for gravitational torques on the robot.

Example 1 (Feasibility Constraints)**.** Consider a simple 2-joint arm robot attempting to move a single sphere object in Figure 2. We illustrate the trust region, the motion set, and the contact impulse set with $\delta q = 0$ and $\|\delta u\|_2 \leq 0.2$; as $\delta q = 0$, this set attempts to approximate the 1-step reachable set at some configuration q, with $\bar{u} = q^{\mathrm{a}}$. We make the following observations from Figure 2.

- The above illustrated configurations are all almost in contact, but still has a positive signed distance (not touching). Consequently, the gradients under true dynamics will all evaluate to zero, leading to a point set. This also signifies the importance of smoothing dynamics to get more informative gradients, and corroborates previous findings [9], [11], [12], [18].
- The reachable set with the linearized model is not sufficient to capture the unilateral nature of contact for all of the examples, as ellipsoids are bi-directional.
- For Configuration A, the primal feasibility constraints lead to a bad approximation of the true object motion set; as a result, imposing both the primal and dual constraints would lead to a small set.
- Only imposing the dual constraints give the best approximation of the object motion set for Configuration A; however, if we add an obstacle in Configuration B, only imposing the dual constraints does not change the set, leading to an insufficient approximation. This can be remedied by adding primal feasibility constraints between the unactuated geometries and environment geometries.
- None of the constraints are informative for approximating the robot motion set, as robots are directly actuated, and the map is close to a multiple of identity.
- From the illustration in Configuration C, we note how our linearized approximation respects singularities, as the sensitivity analysis procedure factor in the configuration of the manipulator.

IV. Local Planning and Control

We now study local gradient-based planning and control, which is one of the core applications of our proposed contact trust region. Among possible formulations of this problem, we focus on coming up with a configuration and input sequence that moves the system towards a goal configuration q_{goal}. The full nonlinear form of this problem can be written as

$$\min_{q_{0:T},u_{0:T-1}} \quad \|q_{\text{goal}} - q_T\|^2_{\mathbf{Q}} + \sum_{t=0}^{T-1} \|u_t - u_{t-1}\|^2_{\mathbf{R}}, \tag{32a}$$

$$\text{s.t.} \quad q_{t+1} = f(q_t, u_t) \quad \forall t, \tag{32b}$$

$$|u_t - u_{t-1}| \leq \eta \quad \forall t, \tag{32c}$$

$$q_0 = \bar{q}_0, \tag{32d}$$

where (32b) enforces dynamics constraints, (32c) enforces input limits (recall that u_t is a position command, thus input limits are enforced in relative form), and (32d) enforces the initial condition.

One of the biggest challenges in solving (32) is handling the non-smooth contact dynamics constraint (32b). Although MIP-based methods [28], [29] have struggled to scale up to complex, contact-rich problems, the MIP formulation reveals why the problem is hard: the search through the exponentially many contact modes.

In this section, we present a method for solving (32) by incorporating contact dynamics smoothing and the contact trust region into a standard trajectory optimization scheme. Through two toy problems, we show that the proposed method can iteratively approach an advantageous contact mode for reaching the given goals, even when the initial guess is in a contact mode with non-informative gradient. In addition, we present how the method can be used as a model predictive controller and be extended to control contact forces.

A. Trajectory Optimization with CTR

Consider a trajectory optimization scheme, where we first obtain some guess of the nominal input $\bar{u}_{0:T-1}$, then roll it out under the dynamics to get the nominal configuration trajectory $\bar{q}_{0:T}$. Under a local approximation of (32) around this nominal trajectory $(\bar{q}_{0:T}, \bar{u}_{0:T-1})$ utilizing gradients of smoothed dynamics and CTR, we search for optimal perturbations $(\delta q_{0:T}, \delta u_{0:T-1})$ by solving the following local trajectory optimization problem with linear dynamics constraints:

$$\mathbf{SubTrajOpt}(\bar{q}_{0:T}, \bar{u}_{0:T-1}, q_{\text{goal}}) = \delta u^\star_{0:T-1}, \text{where} \tag{33a}$$

$$\min_{\delta q_{0:T}, \delta u_{0:T-1}} \quad \|q_{\text{goal}} - q_T\|^2_{\mathbf{Q}} + \sum_{t=0}^{T-1} \|u_t - u_{t-1}\|^2_{\mathbf{R}}, \tag{33b}$$

$$\text{s.t. } (\delta q_t, \delta u_t) \in \mathcal{S}_{\mathbf{\Sigma},\kappa}(\bar{q}_t, \bar{u}_t),\ t = 0 \dots T-1, \tag{33c}$$

$$\delta q_{t+1} = \mathbf{A}_{\kappa,t}\delta q_t + \mathbf{B}_{\kappa,t}\delta u_t,\ t = 0 \dots T-1, \tag{33d}$$

$$q_t = \bar{q}_t + \delta q_t,\ t = 0 \dots T, \tag{33e}$$

$$u_t = \bar{u}_t + \delta u_t,\ t = 0 \dots T-1, \tag{33f}$$

$$|u_t - u_{t-1}| \leq \eta,\ t = 1 \dots T-1, \tag{33g}$$

$$\delta q_0 = 0, \tag{33h}$$

Here $\mathcal{S}_{\mathbf{\Sigma},\kappa}$ in (33c) is the CTR in (22); (33d) is the standard linear dynamics constraint in linear MPC; (33c) and (33d) constitute our local approximation of the contact dynamics constraint (32b); (33h) is due to the initial condition constraint (32d). The sub trajectory optimization problem (33) is a standard SOCP that can be solved by off-the-shelf conic solvers. We note that other constraints as joint limits and torque limits in Section III-F can be added trivially to this formulation by incorporating them into $\mathcal{S}_{\mathbf{\Sigma},\kappa}$.

After obtaining the optimal values $\delta q^\star_t, \delta u^\star_t$, we update the nominal input trajectory with $\bar{u}_t \leftarrow \bar{u}_t + \delta u^\star_t$, and repeat the process of rolling out to obtain nominal configuration trajectory, and searching for local improvements. This iterative scheme is summarized in Algorithm 1.

Algorithm 1: CTR Trajectory Optimization (**CtrTrajOpt**)

Input: Initial state q_0, goal state q_{goal}, input trajectory guess $\bar{u}_{0:T-1}$, iterations limit $n_{\max}$;
Output: Optimized input trajectory $\bar{u}^\star_{0:T-1}$;
$n \leftarrow 0$;
while *not converged* and $n < n_{\max}$ **do**
  $\bar{q}_{0:T} \leftarrow$ Rollout f from q_0 with $\bar{u}_{0:T-1}$;
  $\delta u^\star_{0:T-1} \leftarrow \mathbf{SubTrajOpt}(\bar{q}_{0:T}, \bar{u}_{0:T-1})$;
  $\bar{u}_t \leftarrow \bar{u}_t + \delta u^\star_t \quad \forall t$;
  $n \leftarrow n + 1$
return $\bar{u}_{0:T-1}$

B. Initial Guess Heuristic

As the trajectory optimization problem (32) is nonconvex, the solution that Algorithm 1 converges to can be sensitive to

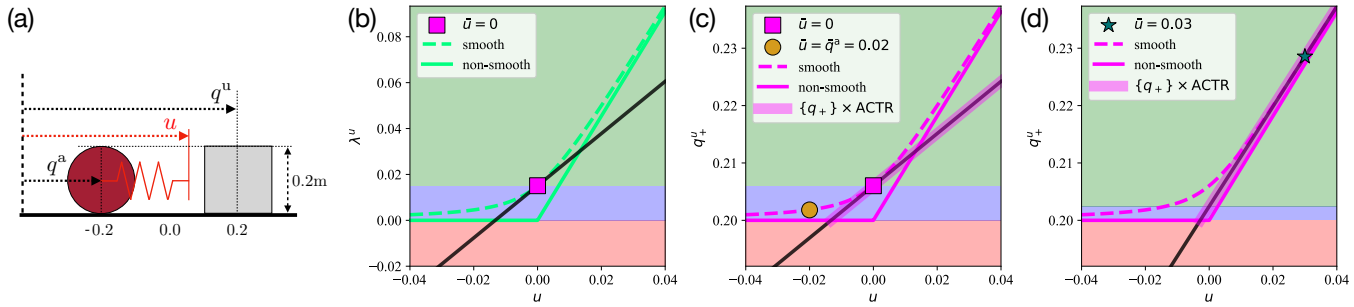


Fig. 3: (**a**) A schematic of the system in Example 2. (**b**) Contact impulse under the true non-smooth dynamics (solid line), and the smoothed dynamics (dashed line) for $q^{\mathrm{a}} = -0.02$ and $q^{\mathrm{u}} = 0.2$. The dynamics is smoothed with $\kappa = 1000$. The magenta box denotes $(\bar{u}, \bar{\lambda}^{\mathrm{u}})$ with $\bar{u} = 0$. The linear model at this configuration is denoted with a black straight line. Note that the dual feasibility constraint requires $\bar{\lambda}^{\mathrm{u}}$ to be positive, which disqualifies the red zone. (**c**) Similar plot for the the next object configuration, q_+^{u}. The magenta box denotes $(\bar{u}, \bar{q}_+^{\mathrm{u}})$ with $\bar{u} = 0$. The magenta translucent band around surrounding the black line represents $\{q_+\} \times$ ACTR, where $\{q_+\}$ is a shorthand notation for the action-only motion set $\mathcal{M}^{\mathrm{u}}_{\Sigma U}(\bar{q}, \bar{u})$. The green, blue, red colored zones denote zones of $q^{\mathrm{u}}_{\mathrm{goal}}$ where the inverse dynamics controller displays similar behavior. (**d**) Similar plot for $\bar{u} = 0.03$.

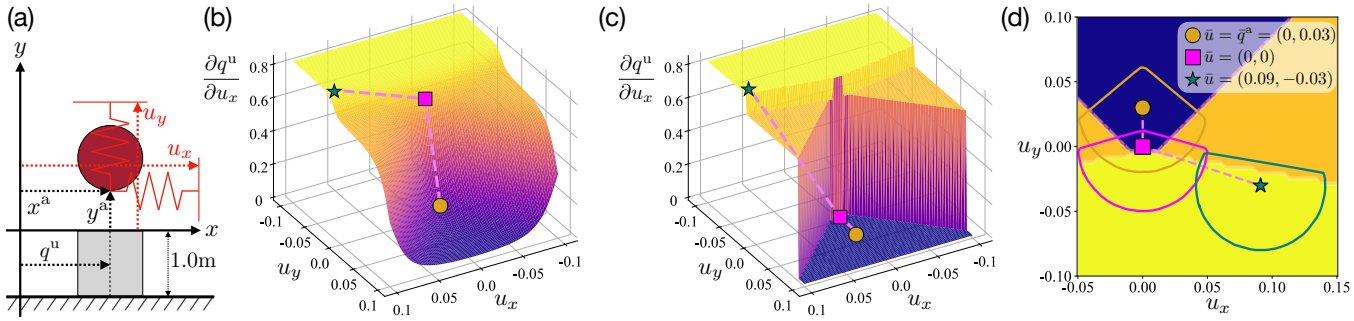


Fig. 4: (**a**) A schematic of the system in Example 3. (**b**) Gradient landscape of the smoothed dynamics at $\bar{q}^{\mathrm{a}} = (0, 0.03)$ and $\bar{q}^{\mathrm{u}} = 0$. The dynamics is smoothed with $\kappa = 1000$. (**c**) Gradient landscape of the non-smooth dynamics at the same nominal configuration. The blue region at the bottom corresponds to the separation contact mode, the two yellow regions the sliding modes, and the yellow region the sticking mode. (**d**) ACTR for different $\bar{u}$'s, overlaid on the non-smooth gradient landscape in (**c**).

the choice of initial guesses for the input trajectory $\bar{u}_{0:T-1}$. The most uninformed initial guess would be to set the position command to be the current configuration of actuated bodies ($\bar{u}_t = \bar{q}^{\mathrm{a}}_0, \forall t$). However, this can be problematic especially when the robots are not in contact with the object in q_0, as $\mathbf{B}^{\mathrm{u}}_\kappa$, the gradient of object motion w.r.t. robot actions, can get close to 0 even with dynamics smoothing when the distance between them are far away.

To alleviate this problem, we can compute a position command that would result in the closest contacting configuration from q_0, and pass this command as the initial guess for $\bar{u}_{0:T-1}$.

To compute this initial guess, we utilize the intuition that the log-barrier smoothed dynamics in (10) results in a force-from-a-distance like effect that pushes away objects even when not in contact. We simply reverse this force field by sending the following torque to the robot,

$$\tau = -\sum_i \mathbf{J}_i^{\mathrm{u}\top} \lambda_i \tag{34}$$

and simulate forward using Drake [53] until the robot is in contact with the object.

C. Examples

To elucidate the role of CTR in Algorithm 1, we show how a single-horizon ($T = 1$) trajectory optimizer solves two toy problems in Example 2 and Example 3. We note that the CTR reduces to the ACTR (Definition 4) when $T = 1$ as the initial condition constraint (33h) eliminates the perturbations on q.

Example 2 (1D Pushing)**.** Consider the 1D system in Figure 3 with two bodies of width 0.2, one actuated (red sphere) and one unactuated (grey box), both constrained to slide on a frictionless surface along the x axis. Let $q^{\mathrm{a}}_0 = -0.02$ and $q^{\mathrm{u}}_0 = 0.2$: the ball is 0.02m away from touching the box. Even with the help of dynamics smoothing, the slope at this q with $\bar{u} = q^{\mathrm{a}}_0$ (yellow circle in Figure 3c) is still small, which is not informative for planning.

In contrast, applying the initial guess heuristics in Section IV-B brings the robot into contact with the object, giving $\bar{u} = 0.0$ (magenta square in Figure 3c). The new $\bar{u}$ creates the linearized dynamics (black line) in Figure 3c, whose slope gets closer to that of the "in contact" piece of the non-smooth contact dynamics. The dual feasibility constraints (22e) removes the region where the linearized contact impulse (black line in Figure 3b) is negative, leading to the ACTR (projection of the magenta translucent line onto the u axis) in Figure 3c.

However, the slope at $\bar{u} = 0$ is still quite different from the slope of the true, non-smooth dynamics. Under this inaccurate linear approximation of the true dynamics, we further analyze the behavior of the optimizer depending on the given goal configuration $q^{\mathrm{u}}_{\mathrm{goal}}$:

- Green zone: If $q^{\mathrm{u}}_{\mathrm{goal}}$ is in the green zone, the optimizer will move in a positive u direction, correctly moving the box towards the goal. However, as the slope is smaller than the slope of the non-smooth dynamics, the commanded u will overshoot $q^{\mathrm{u}}_{\mathrm{goal}}$.

- Blue zone: If $q^{\mathrm{u}}_{\mathrm{goal}}$ is in the blue zone, then the optimizer will move backwards even though the optimal action is to move forward. This is due to the non-physical behavior caused by smoothing: according to the smoothed linearization, when $u = 0$ is commanded, the object will be pushed away by the barrier forces to a location further than $q^{\mathrm{u}}_{\mathrm{goal}}$. To lessen this effect, the optimizer chooses to move backwards.
- Red zone: If $q^{\mathrm{u}}_{\mathrm{goal}}$ is in the red zone, the optimizer would move backwards but stop at the boundary of the red region, which corresponds to the leftmost point in the ACTR in Figure 3c. Without the dual feasibility constraint (22e), a naive linear model with an ETR will move backwards into the red region to attempt pulling the object.

The analysis above suggests that the $\bar{u}$ improved by the initial guess heuristics is still not ideal for planning: as the $\bar{u}$ is located at the boundary between two contact modes which have distinct gradients, a single linear model at $\bar{u}$ approximates neither mode well.

Interestingly, we observe that Algorithm 1 can iteratively improve the local linear approximation under the smoothed dynamics. Consider reaching the goal $q^{\mathrm{u}}_{\mathrm{goal}} = 0.22$ from $q^{\mathrm{a}}_0 = -0.02$, $q^{\mathrm{u}}_0 = 0.2$ and $\bar{u} = 0$. At the first iteration, (33) is solved with the ACTR from Figure 3c. Due to overshooting caused by underestimating the slope of the true dynamics, $\bar{u} \leftarrow 0.03$. At the second iteration, the optimizer constructs the ACTR for $\bar{u} = 0.03$, which is shown in Figure 3d. As $\bar{u} = 0.03$ is deeper in penetration, the ACTR aligns more closely with $\{u | u \geq 0\}$, the domain of the in-contact piece of the non-smooth dynamics. Moreover, the blue zone under the new ACTR, representing the non-physical artifacts caused by smoothing, also shrinks. Under this better local approximation of the non-smooth dynamics, the second iteration returns $\bar{u} = 0.0202$, getting q^{u}_+ much closer to $q^{\mathrm{u}}_{\mathrm{goal}} = 0.22$.

Example 3 (2D Ball Pushing 1D Box)**.** To further illustrate how Algorithm 1 iteratively improves the local linear approximation of the non-smooth contact dynamics, we study a slightly more complex system shown in Figure 4a. The system consists of an unactuated box constrained to slide frictionlessly along the x axis, and an actuated ball that can move in the xy plane and make frictional contact with the box's top surface.

The non-smooth contact dynamics of this system consists of 4 contact modes: sticking, sliding left, sliding right, and separation. Each mode can be identified with an affine piece in the Piecewise-Affine (PWA) contact dynamics, which is continuous but has discontinuous gradients. Specifically, we examine $\partial q^{\mathrm{u}}_+ / \partial u_x$, the component of $\mathbf{B}^{\mathrm{u}}_\kappa = \partial q^{\mathrm{u}}_+ / \partial u$ that describes how much the object(box) moves along the x axis relative to how much the ball(robot) moves along the same axis. As shown in Figure 4c, $\partial q^{\mathrm{u}}_+ / \partial u_x$ is constant within each mode, but is separated from nearby modes by cliffs. The gradient can be made continuous with dynamics smoothing, which is shown in Figure 4b.

Consider the task of moving the box to $q^{\mathrm{u}}_{\mathrm{goal}} = 0.2$ from the initial configuration $q^{\mathrm{u}}_0 = 0$ and $q^{\mathrm{a}}_0 = (x^{\mathrm{a}}_0, y^{\mathrm{a}}_0) = (0, 0.03)$: the ball hovering $0.03\mathrm{m}$ above the box. Starting with $\bar{u} = q^{\mathrm{a}}_0$ puts the system in the "separation" mode, but accomplishing the task requires the ball to push down and drag the object to the right with friction, which is best done in the "sticking" mode.

We show that our method is able to iteratively nudge $\bar{u}$ to the "sticking" mode. Similar to Example 2, we first apply the initial guess heuristics, bringing $\bar{u}$ from q^{a}_0 (yellow circle in Figure 4) to $(0, 0)$ (magenta square). Looking at the smoothed gradient landscape in Figure 4b, we see that the smoothed $\partial q^{\mathrm{u}}_+ / \partial u_x$ climbs up a lot from $\bar{u} = q^{\mathrm{a}}_0$ to $\bar{u} = (0, 0)$. Furthermore, after running Algorithm 1 for 2 iterations, the $\bar{u}$ (green star) reaches the yellow plateau. This sequence of $\bar{u}$'s is plotted in Figure 4b.

Plotting the same $\bar{u}$ sequence over the non-smooth gradient landscape in Figure 4c reveals that $\bar{u}$ jumps from the "separation" mode at the bottom to the "sticking" mode at the top. Without smoothing, mode switches can also be achieved by encoding modes with integer variables and solving the resulting MIP. However, for systems with more contact modes, this approach becomes prohibitively expensive.

Lastly, we plot the ACTR for the above sequence of $\bar{u}$ in Figure 4d. Here we use $\mathbf{\Sigma} = \mathrm{diag}(0.05^{-2}, 0.05^{-2})$, which bounds δu in a circle of radius 0.05. The circular portion of the ACTR boundaries arises from the ellipsoidal constraint (22a), whereas the straight portion follows from the dual feasibility constraint (22e). Compared with the ACTR at q^{a}_0 (dark yellow), the ACTR after applying the initial guess heuristics (magenta) approximates the sticking region much better. Furthermore, as the $\bar{u}$ after two iterations of (1) (dark green) gets deeper into the "sticking" mode, the straight edge of its ACTR aligns more closely with the boundary between sticking and sliding.

D. Model Predictive Control

The above planning algorithm can be readily turned to a controller through Model Predictive Control (MPC). As shown in Algorithm 2, we obtain the optimal trajectory $u^\star_t$ by solving (32) with Algorithm 1 (Line 9), only use the first input $u^\star_0$ to deploy on the true CQDC dynamics Line 10, then re-plan after observing the next configuration. Moreover, we initialize $\mathbf{CtrTrajOpt}$ from the initial guess heuristics (Section IV-B) at the first iteration (Line 6), and from the solution of the previous iteration at later iterations (Line 8).

As done in the iMPC algorithm in [11], [18], we can also utilize the rollouts of an MPC controller as a closed-loop trajectory optimizer similar to how iLQR [66] performs closed-loop planning by rolling out with linear feedback laws.

We further analyze the behavior of MPC on an example with many contact modes, and show the scalability of our method in Example 4. This example also illustrate the importance of the dual feasibility constraints (22e) by a quantitative comparison against the ellipsoidal trust region (19).

Example 4 (Bimanual iiwa)**.** Consider the planar bimanual iiwa system where each iiwa arm has 4 of the 7 available DOFs locked to constrain its motion to the xy plane. The system is tasked with moving a cylindrical object to a desired location. From the initial configuration $q^{\mathrm{u}}_0 = (q^{\mathrm{u}}_x, q^{\mathrm{u}}_y, q^{\mathrm{u}}_\theta) = (0.65, 0.0, 0.0)$, we command a relative large rotation to reach $q^{\mathrm{u}}_{\mathrm{goal}} = (0.65, 0.1, 5\pi/6)$. The controller is rolled out in closed-loop for 35 steps, and the resulting trajectories are displayed in Figure 5.

Algorithm 2: MPC Rollout (**MPC**)

Input: Initial state q_0, goal state q_{goal}, planning horizon T, iterations limit $n_{\max}$, MPC rollout horizon H;
Output: Lists of visited states L_q, applied inputs L_u;
$L_q \leftarrow [q_0]$, $L_u \leftarrow []$;
for $t = 0$ *to* $H-1$ **do**
  if $t == 0$ **then**
    $\bar{u}_{0:T-1} \leftarrow$ Apply initial guess heuristics to q_t^{a};
  else
    $\bar{u}_{0:T-1} \leftarrow$ Initialize from the previous $u^\star_{0:T-1}$;
  $u^\star_{0:T-1} \leftarrow \mathbf{CtrTrajOpt}(q_t, q_{\text{goal}}, \bar{u}_{0:T-1}, n_{\max})$;
  $q_{t+1} = f(q_t, u^\star_0)$;
  L_q.append(q_{t+1}), L_u.append$(u^\star_0)$;
return L_q, L_u

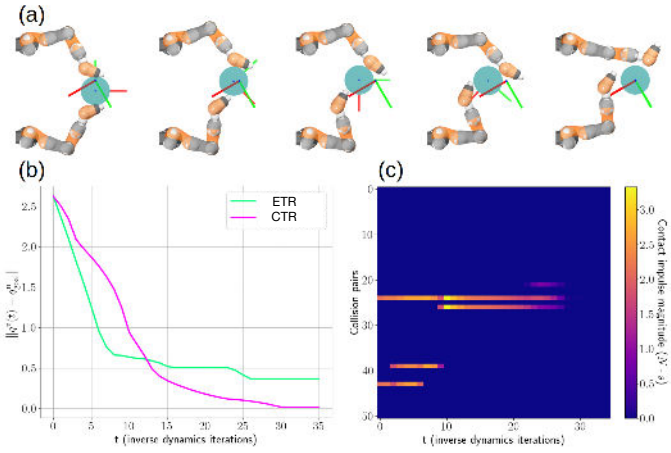


Fig. 5: Visualization for Example 4. (**a**) Visualization of key frames of the closed-loop sequence, with the current q^{u} and the goal $q^{\text{u}}_{\text{goal}}$ displayed as frames. (**b**) Distance to goal for each rollout, $\|q^{\text{u}}_t - q^{\text{u}}_{\text{goal}}\|$. (**c**) Visualization of contact forces of every potential collision pair in the planar iiwa bimanual system.

The rollouts from Figure 5b tell us that the MPC controller indeed benefits from our Contact Trust Region, as it is able to successfully rotate the cylinder by $5/6\pi$ while the Ellipsoidal Trust Region fails to achieve the task. In addition, observing the contact forces in Figure 5c tells us that the controller is able to scalably search across local changes in contact modes and arrive at a different contact sequence than the initial configuration.

E. Force Control

Having a local linear model of the dual variables (contact impulses) also allows us to explicitly control the desired contact forces that we can apply on the object, as we can add the cost

$$\|\lambda^{\text{u}}_i - \lambda^{\text{u}}_{\text{goal}}\|^2_{\mathbf{L}} \tag{35}$$

to optimize how much contact force we apply between a the collision pair indexed by i. Note that when the matrices $\mathbf{Q}$ (in (32a)) and $\mathbf{L}$ have orthogonal eigenspaces, we can apply force in some directions while causing motion in other directions, performing hybrid force velocity control [38], [67]. We illustrate the performance of this force controller in Example 5.

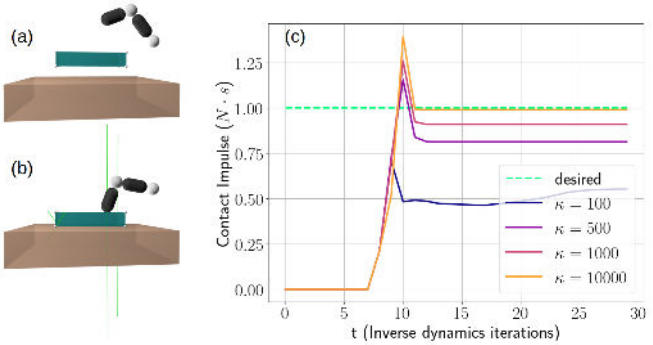


Fig. 6: Visualization for Example 5. (a) Initial configuration at $t = 0$. (b) Final configuration at $t = 30$. The green arrows visualize contact forces applied at spherical collision geometries on the corners of the box. (c) Response of the contact impulses per inverse dynamics iteration for different smoothing parameters.

Example 5 (Force Control)**.** Consider a task where a 2-dof manipulator must be commanded to apply a specific wrench $w^{\text{u}}_{\text{goal}}$ between a box and a table, visualized in Figure 6. As we need to command a wrench between two bodies but our decision variables are contact impulses for each collision geometry pairs, we convert the contact impulses to the net wrench experienced by the body. This can be done by modifying the cost in (35) as

$$\|w^{\text{u}}_{\text{goal}} - \textstyle\sum_{i \in \mathcal{I}_{\text{ue}}} \mathbf{J}^\top_{\text{u}_i} \lambda^{\text{u}}_i\|^2, \tag{36}$$

The result of rolling out the inverse dynamics controller is visualized in Figure 6.

We observe that the controller successfully achieves the desired wrench $w^{\text{u}}_{\text{goal}}$, but more smoothing results in a bias in the achieved wrench due to the fact that the controller attempts to achieve $w^{\text{u}}_{\text{goal}}$ under smoothed dynamics, and sends a less penetrating position command as a result.

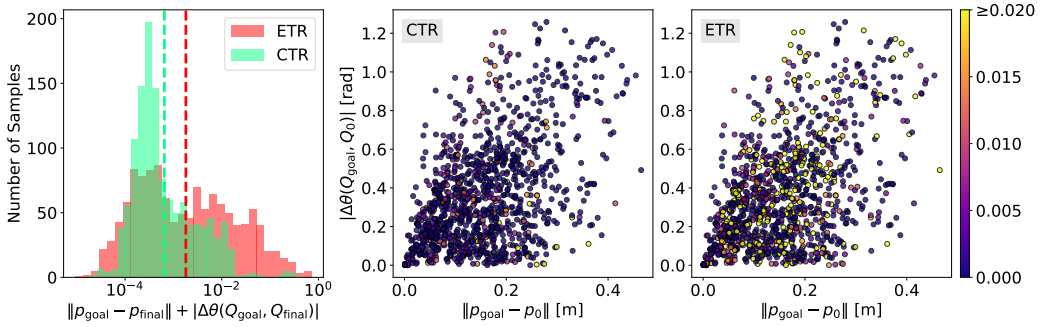
V. Local Planning Experiments

In this section, we ask whether the planning method presented in Section IV is successful under the CQDC dynamics used as dynamics constraints in planning. We are particularly interested in answering the following questions:

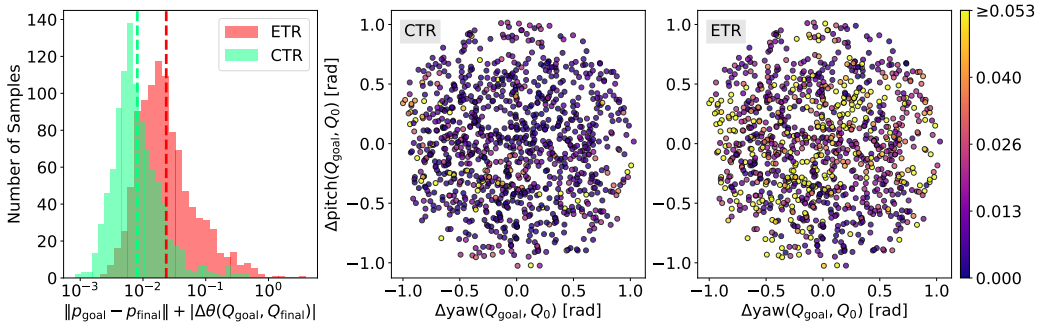
- Does the CTR improve performance over the ETR?
- How does the planning horizon T in (32) affect MPC errors?
- Does the method successfully reach a diverse set of goals on complex systems such as dexterous hands?

To answer these questions and demonstrate the scalability of our method, we conduct statistical analysis on two contact-rich robotic systems:

- the planar `IiwaBimanual` system in Example 4, which comprises of 3 unactauted DOFs, 6 actuated DOFs and 29 collision geometries. The whole system is constrained to the xy plane, with gravity pointing along the negative z direction. The bucket measures 0.28m in diameter. The task is to rotate the object, a cylindrical bucket, to target $SE(2)$ poses.
- the 3D `AllegroHand` system, which comprises 6 unactauted DOFs, 16 actuated DOFs and 39 collision geometries. The object, a 6cm cube, is unconstrained. The hand's wrist is welded to the world frame. The task is to reorient the cube to target $SE(3)$ poses.



(a) Iiwa Bimanual



(b) Allegro Hand

Fig. 7: Results for running the CTR and ETR variants of Algorithm 2 on the pairs of goals and initial conditions generated in Section V-A1. The histogram shows how many samples reached a specified error in the x axis, where error is defined as $\|p_{\text{goal}} - p_{\text{final}}\| + |\Delta\theta(Q_{\text{goal}}, Q_{\text{final}})|$. The dotted lines in the histogram represent the average errors. Each dot in the scatter plot corresponds to a pair of an initial condition q_0^{u} and a goal $q_{\text{goal}}^{\text{u}}$. Color of the dots is determined by the error. The two scatter plots in the same row share the same color scale, whose maximum value is the average of the errors for reaching the goals with ETR. (**a**) For IiwaBimanual, the x axis of the scatter plots is the position difference between q_0^{u} and $q_{\text{goal}}^{\text{u}}$, the y axis the angular difference. (**b**) For AllegroHand, as the position difference is small for all goals, the x axis shows the yaw angle difference between q_0^{u} and $q_{\text{goal}}^{\text{u}}$, and the y axis the pitch angle difference. The roll, pitch and yaw axes are defined in Figure 12a.

A. Experiment Setup

1) Goal Selection : When statistically evaluating our method, the success rate depends on *which goals* are commanded from *which initial configurations*. For local optimization methods, picking the pairs of initial configurations and goals is nontrivial: we cannot pick goals that are too easy, but we also cannot pick goals that can only be reached with highly non-local movements and extensive exploration. These hard goals do not help us evaluate the performance of our planner, as we are more concerned with the ability to stabilize locally.

To give goals that are considered more locally reachable yet far enough to be challenging for the planner, we utilize the following schemes. For the iiwa bimanual environment, we first prescribe some initial state $\bar{q}$ by sampling contact points on the surface of the object and solving inverse kinematics; then, the goals are sampled from the boundary of the object motion set $\mathcal{M}^{\text{u}}_{\mathbf{\Sigma}U,\kappa}(\bar{q}, \bar{u})$ with $\bar{u} = \bar{q}^{\text{a}}$. However, we set the ellipsoid radius to be large so that the task is sufficiently challenging. The resulting goals have maximum 120° rotation and $0.4m$ translation from the initial state (Figure 7). Similarly, we prescribe some grasp for the Allegro hand and sample orientations that have about a 60° difference from the initial state according to the axis-angle metric.

With the above procedure, we sampled 1233 pairs of initial states and goals for the iiwa bimanual system and 1000 pairs for the allegro hand system. The sampled pairs are shown in the scatter plots in Figure 7, which exhibit a broad distribution, indicating a high level of diversity in the samples. Moreover, as we only care about moving the object to desired configurations, we do not generate robot goal configurations.

2) Evaluation Metrics: We evaluate the performance of our method by comparing the difference between $q_{\text{final}}^{\text{u}}$, the final object configuration reached by the by Algorithm 2, and $q_{\text{goal}}^{\text{u}}$, the goal object configuration. As we are only concerned with reaching object goals, we set the robot-related entries in the cost matrix $\mathbf{Q}$ in (33) to 0.

We split the object configuration q^{u} into a quaternion Q and a position p: $q^{\text{u}} := (Q, p)$, both expressed relative to the world frame. We divide the error in q^{u} into a *translation error* $\|p_{\text{goal}} - p_{\text{final}}\|$, and an *rotation error* $\Delta\theta(Q_{\text{goal}}, Q_{\text{final}})$, where $\Delta\theta(\cdot, \cdot)$ returns the angular difference between two unit quaternions.

As a reference, we summarize in Table II the average amount of translation and rotation between the initial and goal object configurations generated in Section V-A1. These would be the object pose tracking errors if the planning algorithm being evaluated did nothing.

	IiwaBimanual	AllegroHand
Mean $\|p_{\text{goal}} - p_0\|$ [mm]	152	16
Mean $\|\Delta\theta(Q_{\text{goal}}, Q_0)\|$ [mrad]	356	788

TABLE II: Mean translation and rotation differences for pairs of initial and goal object states generated in Section V-A1.

3) Implementation Details: The numerical experiments are run on a Macbook Pro with the M2 Max chip and 64GB of

RAM. We use the same open-sourced implementation of the CQDC dyanmics as in [11]. The convex subproblem (33) in Algorithm 1 is solved with Mosek [68].

B. Effect of Planning Horizon T

First, we study how the planning horizon T used in (32) affects the MPC errors. Specifically, for the IiwaBimanual system and the corresponding goals selected from Section V-A1, we run Algorithm 2 with planning horizon from 1 to 5, a maximum iteration of 2 ($T = 1 \dots 5$, $n_{\max} = 2$ in Algorithm 1), and a MPC rollout horizon H of 10.

The results are summarized in Table III. Surprisingly, we found that the planning horizon T has little to no effect on the final error for MPC: our method is able to reach the goals within a tight tolerance for all values of T. We believe that this is due to two reasons: i) we have chosen the initial configurations and goals to be reasonably reachable using our trust region, and ii) the CQDC model allows large change in configurations within a single step, so we are able to get away with significantly less horizon compared to other MPC tasks such as dynamic locomotion.

	T = 1	2	3	4	5
Translation (mm)	2.02	1.99	1.64	2.11	2.1
Rotation (mrad)	2.72	3.51	3.25	4.13	5.03

TABLE III: Mean translation and rotation Errors for IiwaBimanual under different planning horizon T.

C. Effect of Trust Region Radius

Next, we investigate whether the additional feasibility constraints in CTR help us better solve the MPC problem in Section IV-D. In Section III, we hypothesized that a large ellipsoidal region would result in violations of important contact constraints such as not allowing contact forces to pull. In addition, we hypothesized that if one were to search for an ellipsoid that is fully inscribed within the feasible set, it would result in an overly small trust region that would require more iterations.

Thus, we could ask: how does the trust region radius $\mathbf{\Sigma}$ affect the performance gap between CTR and the ellipsoidal region? As the radius shrinks, we would expect there to be less of a difference in performance as the ellipsoidal trust region would also not violate feasibility constraints, simply due to being small in size.

To test this, we run Algorithm 2 on the IiwaBimanual system with different trust region radii. We set $\mathbf{\Sigma}$ to a constant scaling of the identity matrix, i.e. $\mathbf{\Sigma} = r\mathbf{I}$ where $r \in \mathbb{R}_+$ is the radius. We also use $T = 1$, $n_{\max} = 2$ and an MPC rollout horizon of 10. Lastly, as we decrease the trust region radius r, we increase the number of iterations to keep the number of total traveled steps in the MPC rollouts approximately similar.

The results are illustrated in Figure 8. Interestingly, increasing the trust region radius r improves CTR performance but degrades ETR performance. Furthermore, the performance gap between CTR and ETR decreases as we decrease the trust region radius, as we hypothesized.

Moreover, to experimentally justify our choice of dropping the primal feasibility constraints in CTR (Section III-D), we also ran MPC with FTR (Section III-C), which enforces both primal and dual feasibility constraints. Consistent with the visualization in Figure 2, we observe that adding in primal feasibility constraints decreases performance.

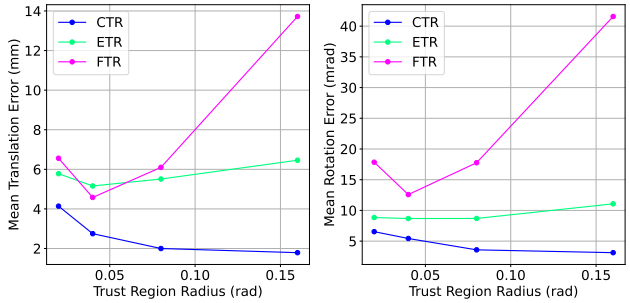


Fig. 8: Comparison of error as trust region radius is varied for the iiwa bimanual example. Experiments with smaller trust region radii are run with more iterations.

D. Can Our Method Reach Goals?

Finally, for all initial condition and goal pairs generated in Section V-A1, we run Algorithm 2 and analyze the results using the metrics in Section V-A2. A subset of the MPC hyperparameters is summarized in Table IV.

	T	$n_{\max}$	r	H
IiwaBimanual	1	2	0.1	10
AllegroHand	1	3	0.05	50

TABLE IV: MPC hyperparameters: T and $n_{\max}$ are defined in Algorithm 1; r is the trust region radius (recall $\mathbf{\Sigma} = r\mathbf{I}$); H is the MPC rollout horizon in Algorithm 2.

As shown in Figure 7, on both IiwaBimanual and AllegroHand, MPC with CTR can reach most goals well, achieving average errors much smaller than the naive ETR. As Figure 7 combines rotation and translation errors, we present them separately in Table V to better illustrate their relative magnitudes.

	IiwaBimanual		AllegroHand	
	Trans. [mm]	Rot. [mrad]	Tran. [mm]	Rot. [mrad]
CTR	1.9	2.0	2.5	12.8
ETR	9.2	10.8	5.5	47.3

TABLE V: Mean translation and rotation errors for the combined errors in Figure 7.

We also record the wall clock time spent on the most time-consuming steps in Algorithm 2, shown in Table VI. **CtrTrajOpt** for AllegroHand is considerably slower due to the higher number of dual feasibility constraints (24e). When constructing the CTR, we only add dual feasibility constraints for robot collision geometries close to the object. For IiwaBimnaual, only a few robot collision geometries are near the bucket at a time, whereas most of the geometries on the Allegro hand are close since the cube is typically grasped.

VI. Stabilization under Second-Order Dynamics

In Section V, we presented results for running MPC on the CQDC dynamics (7). However, although it removes the force-at-a-distance effect from smoothing, the CQDC dynamics is

time unit [ms]	IiwaBimanual	AllegroHand
Per IGH	16.60	40.90
Per **CtrTrajOpt**	4.80	135.94
Per $q_{t+1} = f(q_t, u_0^\star)$	0.86	5.70
Total Time	72.30	7150.23

TABLE VI: Breakdown of the average runtime (wall clock) for Algorithm 2. IGH stands for Initial Guess Heuristic. The total time row is obtained using the parameters in Table IV. Total time $\approx$ IGH time + $H\times$ (time per **CtrTrajOpt** + time per f).

still different from real contact physics: not only are second-order effects ignored, it also introduces a small gap between objects undergoing sliding friction [10, Section IV-A2], which is an artifact known as "hydroplaning" [69, Appendix B] shared by contact dynamics formulations that utilize Anitescu's convex relaxation for contact dynamics [51].

In this section, we would like to understand how the gap between the CQDC and real physics affect MPC performance in more realistic settings. Specifically, we focus on answering the question: can the MPC controller using the CQDC dynamics perform closed-loop stabilization i) on high-fidelity simulated second-order dynamics, and ii) on hardware?

A. Addressing the CQDC to Second-order Dynamics Gap

We first introduce two techniques which have empirically boosted the performance of MPC on second-order dynamics. The modified MPC algorithm is summarized in Algorithm 3.

1) Reducing feedback rate: To handle second-order dynamics, f_{2nd}[2], in MPC, can we simply replace the CQDC dynamics (f in Line 10 of Algorithm 2) with f_{2nd}? Considering the computation time shown in Table IV, we can run the MPC loop at about 20Hz on IiwaBimanual and 5Hz on AllegroHand. However, we observed significant instability when we tried this: the robot oftentimes keeps shaking the object around the starting configuration without making progress towards the goal until timeout.

Interestingly, we found the shaking behavior can be eliminated by reducing the feedback rate. As shown in Algorithm 3, instead of re-planning multiple times per second, we compute a longer trajectory (usually 1-2 seconds) using Algorithm 2 for multiple steps ($H > 1$) (Line 4), roll out the *entire* returned action sequence L_u *open-loop* on the second-order dynamics (Line 5), and repeat.

2) Applying Initial Guess Heuristics More Frequently: When running trajectories generated by Algorithm 2 open-loop on second-order dynamics, we observed that the robot sometimes loses contact with the object. We hypothesized that this is caused by the hydroplaning artifact of the CQDC dynamics, which allows the robot to exert sliding friction forces on the object without actually touching it. As the hydroplaning artifact does not exist for sticking friction [51], we can alleviate hydroplaning by regularly pulling the robot back into contact with object using the initial guess heuristics (Section IV-B). Specifically, in Algorithm 2, instead of only applying the initial guess heuristics at the first iteration (Line 5 to Line 8), we apply it at every iteration to the corresponding q_t^{a}.

[2]With the term "second-order dynamics" and the symbol f_{2nd}, we refer to both simulated second-order dynamics and hardware dynamics.

We denote this modified version of MPC with the symbol $\mathbf{MPC}_{\text{Proj}}$ in Line 4 of Algorithm 3, where the subscript denotes the projection back to the contact manifold, which is the effect of applying the initial guess heuristics.

Algorithm 3: MPC with 2nd-order Dynamics f_{2nd}

1 **Input:** Initial state q_0, goal state q_{goal}, planning horizon T, iterations limit $n_{\max}$, MPC rollout horizon H, number of re-plans N;
2 $i \leftarrow 0$;
3 **while** q_{goal} *not reached* and $i < N$ **do**
4 $\quad q^\star_{0:H}, u^\star_{0:H-1} \leftarrow \mathbf{MPC}_{\text{Proj}}(q_0, q_{\text{goal}}, T, n_{\max}, H)$;
5 $\quad q_{\text{final}} \leftarrow$ Apply $u^\star_{0:H-1}$ to f_{2nd};
6 $\quad q_0 \leftarrow q_{\text{final}}$;
7 $\quad i \leftarrow i + 1$

B. Experiment Setup

We evaluate Algorithm 3 using the same final-distance-to-goal metric proposed in Section V-A2 on both IiwaBimanual and AllegroHand. For each system, we run two ablation variants of our algorithm, each omitting a crucial component.

- **Closed-loop**. This is running Algorithm 3 as is with multiple re-plans ($N > 1$).
- **No Heuristics**. We replace $\mathbf{MPC}_{\text{proj}}$ with the **MPC** defined in Algorithm 2. In other words, we apply the initial guess heuristics only at the first iteration in MPC, instead of at every iteration.
- **Open-loop**. This can be interpreted as reducing feedback rate to the extreme: there is no re-planning at all. Instead, we plan a long trajectory that gets the object all the way to the goal and run it open-loop.

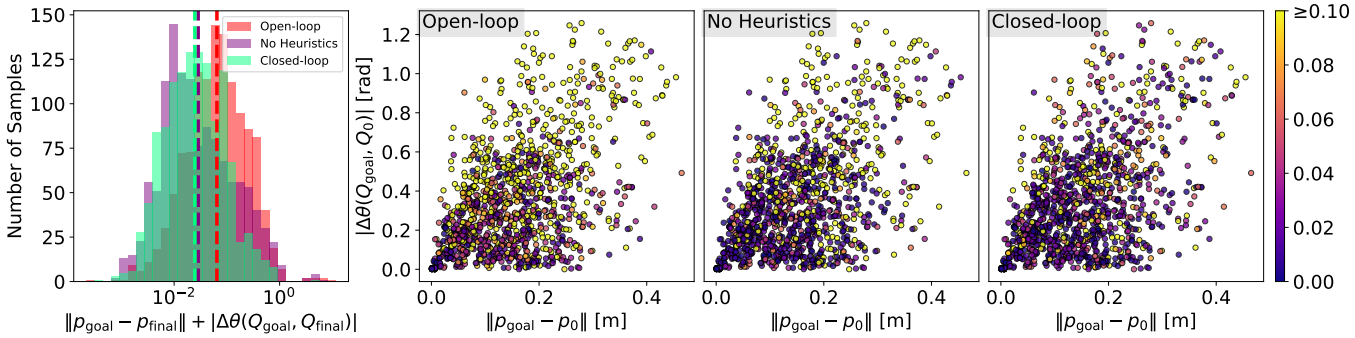
We use the same **CtrTrajOpt** parameters from Table IV. Other important MPC hyperparameters are summarized in Table VII. For closed-loop IiwaBimanual experiments, we use larger N for goals that are further away, while keeping $N \times H$ around 25. In contrast, as slippage happens a lot more often on the AllegroHand system, we keep N constant for all goals to give MPC more chances to recover.

	N (Number of re-plans)		H (MPC rollout horizon)	
System	open-loop	closed-loop	open-loop	closed-loop
IiwaBimanual	1	Variable	25	5
AllegroHand	1	5	50	10

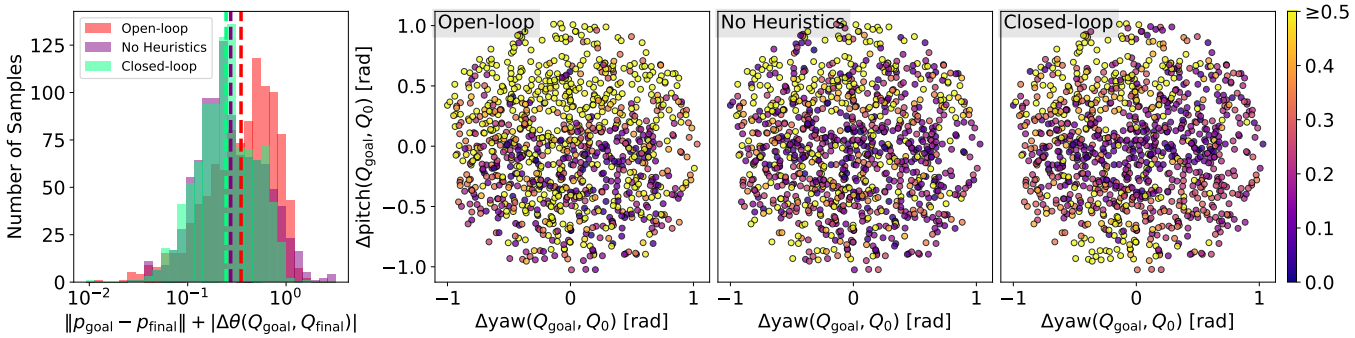
TABLE VII: Hyperparameters for the MPC in Algorithm 3. **No Heurisitcs** and **Closed-loop** share the same parameters.

Simulation Setup. We incorporate the IiwaBimanual and AllegroHand systems presented in Section V into Drake [15], which employs a complete second-order dynamics model along with state-of-the-art contact solvers [53], [70]. We maintain identical collision geometries, robot controller stiffness, and friction coefficients between the CQDC dynamics and Drake.

Hardware Setup. We keep the MPC and physical parameters of the system, such as robot feedback gains, object shape, mass, and friction, consistent between Drake and hardware.



(a) Iiwa Bimanual



(b) Allegro Hand

Fig. 9: Results for running the three variants of Algorithm 3, namely open-loop, no heuristics and close-loop, on the pairs of goals and initial conditions generated in Section V-A1. All figure elements follow the same definitions used in Figure 7.

	IiwaBimanual		AllegroHand	
	Trans.[m]	Rot.[rad]	Trans.[m]	Rot.[rad]
Open-loop Sim	0.048	0.115	0.011	0.443
No Heuristics Sim	0.050	0.066	0.015	0.358
Closed-loop Sim	0.021	0.039	0.014	0.290
Open-loop HW	0.016	0.056	0.014	0.323
Closed-loop HW	0.013	0.024	0.018	0.258

TABLE VIII: Mean translation and rotation errors for simulation (Sim) and hardware (HW) experiments.

We measure the object pose utilizing the OptiTrack motion capture system. Passive spherical reflectors are sufficient for the bucket in IiwaBimanual. For the cube in AllegroHand, in order to not alter the collision geometry with external markers, we embed light-emitting active markers in the cube. In addition, we post-process the cube pose by solving an inverse kinematics problem that projects the raw OptiTrack measurement out of penetration with the Allegro hand.

C. Results & Discussion

We plot the results of our experiments in Figure 9, and display the translation and rotation errors separately in Table VIII. We discuss some of our findings from the experiments.

*1) **Closed-loop** vs. **Open-loop**:* Closed-loop MPC clearly outperforms open-loop on both systems. This suggests that even though reducing the feedback rate (Section VI-A1) is needed for stability, feedback is still necessary to reduce object tracking errors.

2) AllegroHand vs. IiwaBimanual: The average errors from closed-loop MPC is much smaller on IiwaBimanual, which we attribute to the Allegro task's inherently difficulty. As reaching goal poses on the Allegro often requires lifting the cube (Figure 12b), any slip can cause the cube to drop, resetting progress and resulting in large errors. In contrast, on IiwaBimanual, the bucket remains on a tabletop, so slipping merely slows progress without eliminating it.

Challenges due to slipping is also evident in the distribution of tracking errors in the scatter plots in Figure 9b. Under the closed-loop MPC, Allegro's tracking error remains low for large yaw angles when pitch angle is near zero, because the cube's bottom face stays close the palm for such goals. In other goal orientations requiring more lift, the cube is less supported by the palm, increasing the risk of dropping.

*3) **Closed-loop** vs. **No Heuristics**:* Although the initial guess heuristics only slightly reduces the mean error (see the histograms in Figure 9), it significantly lower the incidence of large errors (i.e., those close to or exceeding 1). Such large errors usually happen when the object falls off the supporting surface (Figure 11d and Figure 12d) after contact is lost during MPC. By regularly applying the initial guess heuristics, the robot is constantly pulled back to the object, greatly reducing the chances of losing contact. As demonstrated in Figure 11b and Figure 12c, although lost contact occurred and caused some error, the object did not fall, thanks to the initial guess heuristics.

Moreover, applying the initial guess heuristics results in cleaner, shorter trajectories. Once contact is lost during the execution of Algorithm 3(e.g. Figure 11), the subsequent **MPC** step (Line 4) typically starts by commanding a large robot movement to reestablish contact. However, under the **closed-loop** variant of Algorithm 3 where the initial guess heuristics is applied more frequently, this movement is significantly

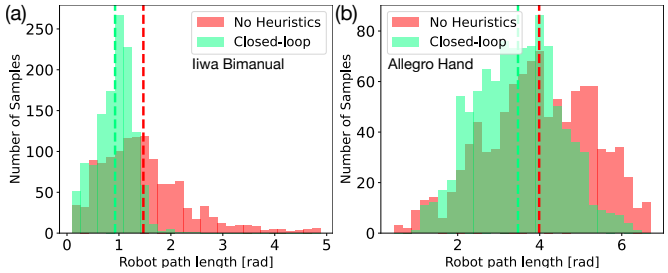


Fig. 10: Robot Path Length Comparison between the **Closed-loop** and **No Heuristics** variants of Algorithm 3. Dotted vertical lines indicate the mean values of each color-coded sample set.

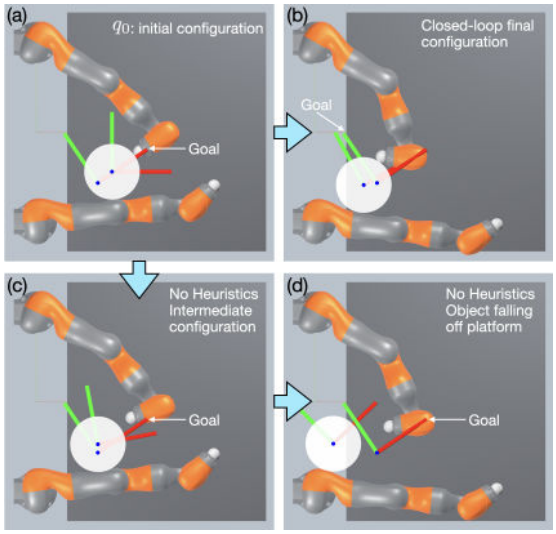


Fig. 11: Failures of Algorithm 3 on iiwabimanual. The dark gray box represents the table top which supports the bucket. The light gray area represents empty space.

smaller.

In fact, for each pair of iniditial and goal configurations, we can compute the path length travelved by the robot as follows:

$$\text{Robot Path Length} = \int_0^{t_{\text{final}}} \|\dot{q}^{\text{a}}(t)\|_2 \, \mathrm{d}t, \qquad (37)$$

where t_{final} is the duration of the experiment and $q^{\text{a}}(t) : \mathbb{R} \to \mathbb{R}^{n_{\text{a}}}$ the robot trajectory. As shown in Figure 10, **closed-loop** MPC results in shorter robot path lengths.

4) Hardware Experiments: Using voxel-grid down-sampling, we get 50 uniformly-spaced pairs of initial states and goals for each system from the pairs used for simulation experiments. Shown in the last two rows of Table VIII, the average errors of hardware experiments are comparable to those of simulation. All hardware and simulation videos are available in an interactive plot on our project website.

VII. Actuator Placement

Through our method presented in Section IV, as well as the results in Section V and Section VI, we have shown how our presented MPC allows successful stabilization to *local* goals.

However, relying solely on finite-horizon MPC formulations can be limiting if the goal is sufficiently "far away". Specifically, goals that are more difficult to reach require the controller to momentarily make suboptimal actions in order

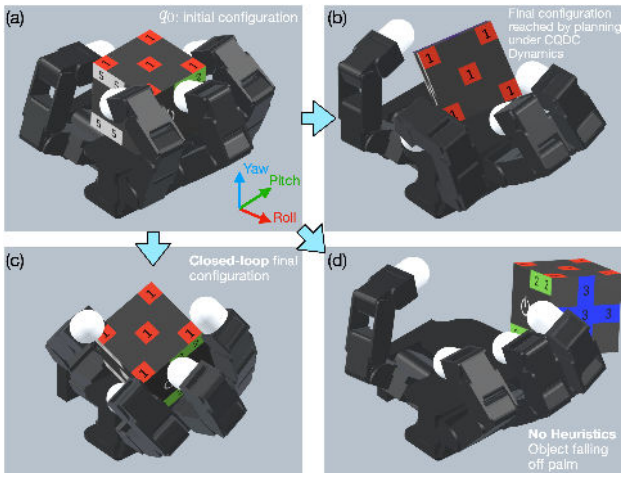


Fig. 12: Failures of Algorithm 3 on AllegroHand.

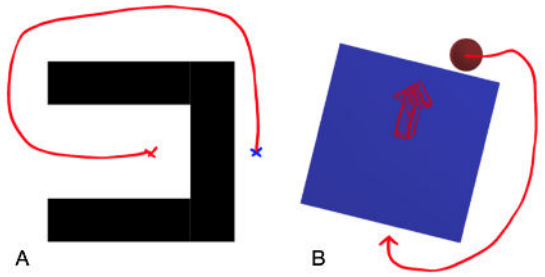


Fig. 13: Difficult cases for greedy finite-horizon MPC that require long-horizon exploration capabilities. A. A classical example for path-finding where the robot needs to go from the red X to the blue X. Note that the robot needs go back first, momentarily suffering an increase in Euclidean distance, before it can proceed towards the blue X and reach the goal. B. A similar planar-pushing example in manipulation, where the goal is to push the box upwards. From this configuration, the agent must break contact and travel all around to the back of the box, causing no effect in making progress towards the goal in terms of box movement, in order to push the box up.

to be optimal in the long run. Such problems inherently cannot be addressed solely by greedy finite-horizon MPC, as illustrated by key examples in Figure 13. To tackle such problems, classical motion planning and RL literature either resort to sophisticated exploration strategies or utilize dynamic programming to estimate a long-horizon value function.

To efficiently conduct long-horizon planning for contact-rich manipulation, we utilize two key insights. The first insight comes from dynamic programming: knowledge of key intermediate states allows us to decompose a difficult long-horizon problem into multiple shorter subproblems in which greedy strategies can be more effective. For the planar-pushing example in Figure 13, consider the configuration in which the ball is at the bottom of the box as opposed to the top. If the planner had i) seen this configuration before and ii) known that this configuration is significantly more advantageous for pushing the box upward using local control, it can attempt to go towards this advantageous configuration first before attempting to establish contact with the box.

Next, we use the fact that because we have full control of the actuated DOFs, it is possible to leverage collision-free motion planning to move from one configuration to another

without reasoning about contact dynamics (provided that a feasible path exists). This allows us to abstract away the details of collision-free motion planning at when we plan contact interactions, and assume that the robot can teleport from one configuration when the object is at a stable configuration. For instance, in the planar-pushing example in Figure 13, we can freely search for a good pusher configuration to push the box upwards without having to worry about the details of which path the robot has to take.

In this section, we leverage the CTR and our proposed MPC method to first search for these meaningful key configurations, while the next section will show how we can chain these key configurations together.

A. Problem Specification

Given a current configuration of the unactuated object q^{u} and some goal configuration q_g^{u}, we ask the following question: what is a good configuration to place the actuators q^{a}, if we want the resulting full configuration $q = (q^{\mathrm{a}}, q^{\mathrm{u}})$ to be *advantageous* for driving q^{u} to q_g^{u} with local MPC? Formally, we write this optimization problem as

$$\min_{q^{\mathrm{a}}} \; C(q^{\mathrm{a}}; q^{\mathrm{u}}, q_g^{\mathrm{u}}) \tag{38a}$$

$$\text{s.t.} \;\; q_{lb}^{\mathrm{a}} \le q^{\mathrm{a}} \le q_{ub}^{\mathrm{a}}, \tag{38b}$$

$$\phi_i(q^{\mathrm{a}}, q^{\mathrm{u}}) \ge 0 \quad \forall i, \tag{38c}$$

where (38b) are joint-limit constraints, (38c) enforce non-penetration constraints for every collision pair indexed by i, and (38a) is our cost criteria for judging how fit q^{a} is in driving q^{u} to q_g^{u}. Our cost C is consisted of two terms: the finite-horizon value function of the MPC policy (Section VII-A1), and a regularization term for robustness (Section VII-A2).

1) Finite-Horizon Value Function of the MPC Policy: A natural way to query for the fitness of the actuator configuration q^{a} is to utilize the cost of our finite-horizon MPC problem (32a) incurred from the MPC rollout defined in Algorithm 2,

$$V(q^{\mathrm{a}}; q^{\mathrm{u}}, q_g^{\mathrm{u}}) = \|q_g^{\mathrm{u}} - q_T^{\mathrm{u}}\|_{\mathbf{Q}}^2 + \sum_{t=0}^{T-1} \|u_t - u_{t-1}\|_{\mathbf{R}}^2, \tag{39a}$$

$$\text{s.t.} \quad q_{0:T}, u_{0:T-1} = \mathbf{MPC}(q_0, q_g^{\mathrm{u}}, T, n_{\max}, H), \tag{39b}$$

$$q_0 = (q^{\mathrm{a}}, q^{\mathrm{u}}). \tag{39c}$$

We note that although MPC rollout in Algorithm 2 accepts a full configuration q_g as a goal, we can give it unactuated configurations only (q_g^{u}) by setting the cost terms of the actuated objects to zero, $\mathbf{Q}_a = \mathbf{0}$, as we did in Section V-A2. In addition, due to the efficiency of the MPC controller, the finite-hozion value function can be quickly queried online. However, the landscape of this value function, as visualized for a simple problem in Figure 14, is multi-modal with many local minima and maxima. This hints at the necessity of global optimization when we search for the minimizers of (39).

2) Robustness Regularizer: Is the value function in (39) sufficient as a cost? Although it correctly evaluates the fitness of a given q^{a} in terms of closed-loop goal reaching, we found that there may be cases where multiple configurations are equally fit, yet one configuration provides more robustness compared to others.

For example, consider the task in Figure 15 where the task is to push the ball slightly to the right. Both of these configurations are nearly equal in the goal-reaching cost (see Table IX); yet, the configuration in Figure 15a would be preferred in practice, as both fingers can be used to reject small disturbances during closed-loop control.

Configuration	(a)	(b)
MPC Value function V	2.82e-4	2.16e-4
Max inscribed sphere radius.	7.12e-3	1.68e-3

TABLE IX: Comparison of MPC value function cost and maximum inscribed sphere radius for two configurations in Figure 15.

To formalize this notion of robustness, we take inspiration from classical grasping metrics [61], [64], [72], as well as the connection between the CTR and the classical wrench set in Section III-E. Specifically, we adopt a worst-case metric [61] that reasons about the maximum wrench a grasp can resist along any direction. Geometrically, this quantity corresponds to the maximum-inscribed sphere in the wrench set.

Formally, consider a unit vector $v \in \mathbb{R}^{n_{q_{\mathrm{u}}}}$. Then, the radius of the maximum-inscribed sphere within the wrench set can be described as the following mini-max problem,

$$r(q^{\mathrm{a}}; q^{\mathrm{u}}) := \min_{v} \max_{r \in \mathbb{R}} \; r \tag{40a}$$

$$\text{s.t.} \; \|v\| = 1 \tag{40b}$$

$$rv \in \mathcal{W}_{\mathbf{\Sigma U},\kappa}(\bar{q} = (q^{\mathrm{a}}, q^{\mathrm{u}}), \bar{u} = q^{\mathrm{a}}). \tag{40c}$$

To evaluate this quantity, we first make a polytopic approximation of the wrench set $\mathcal{W}_{\mathbf{\Sigma U},\kappa}$ by sampling from the ACTR using a rejection sampling scheme. This scheme i) first samples from the ETR ellipsoid and ii) rejects samples that do not obey feasibility constraints. Then, we fit a convex hull to these samples using the `Qhull` library.

Once we have a polytopic representation of the convex set in H-rep, (i.e. $\{z | \mathbf{a}_{\mathcal{W},i}^\top z + b_{\mathcal{W},i} \le 0 \; \forall i\}$), we can find r by solving a variant of the Chebyshev center problem,

$$\max_{r \ge 0} \; r \tag{41a}$$

$$\text{s.t.} \; \|a_i\|_2 r + b_i \le 0 \quad \forall i. \tag{41b}$$

3) Total Cost: We combine the two costs in Section VII-A1 and Section VII-A2 use a weighting term $\alpha \in \mathbb{R}_{\ge 0}$:

$$C(q^{\mathrm{a}}; q^{\mathrm{u}}, q_g^{\mathrm{u}}) = V(q^{\mathrm{a}}; q^{\mathrm{u}}, q_g^{\mathrm{u}}) - \alpha r(q^{\mathrm{a}}; q^{\mathrm{u}})^2, \tag{42}$$

where we note that the radius is subtracted as it is a reward.

B. Optimization by Monte-Carlo Sampling

Solving (38) is a challenging problem, as i) the gradient of the cost function is quite difficult to obtain [72], and ii) it requires global search, as evidenced by the nonconvex cost landscape in Figure 14. As a result, we resort to a simple sampling-based search which samples from the feasible set of (38) by rejection sampling, then chooses the best sample.

However, due to the high variance of this process in high-dimensional spaces, as well as the complexity of navigating high-dimensional configuration spaces, we found that directly applying this strategy is not very effective beyond simple planar problems. As a result, we introduce a few heuristic changes to make this optimization more tractable.

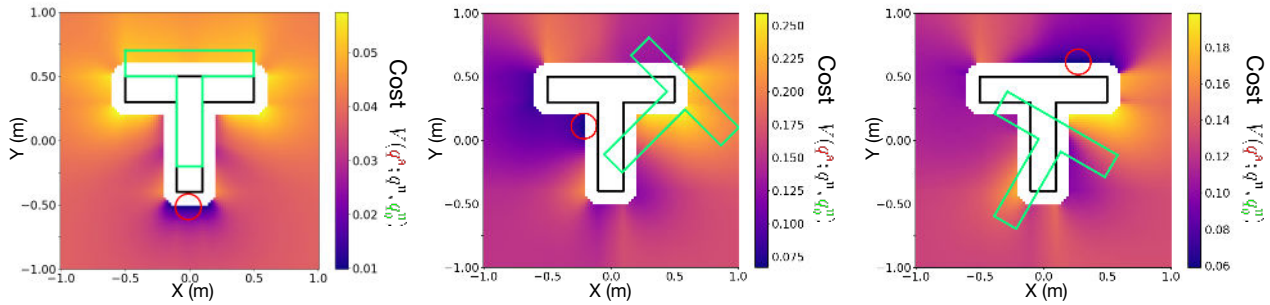


Fig. 14: Visualization of the landscape of (39) for the planar pusher-T example inspired by [71]. The black T corresponds to the current configuration of the object q^{u}, and the green T corresponds to the goal configuration of the object q_g^{u}. The landscape corresponds to the position of the round pusher q^{a}, colored with the cost function $V(q^{\mathrm{a}}; q^{\mathrm{u}}, q_g^{\mathrm{u}})$. The red pusher configuration corresponds to the global optima of the landscape.

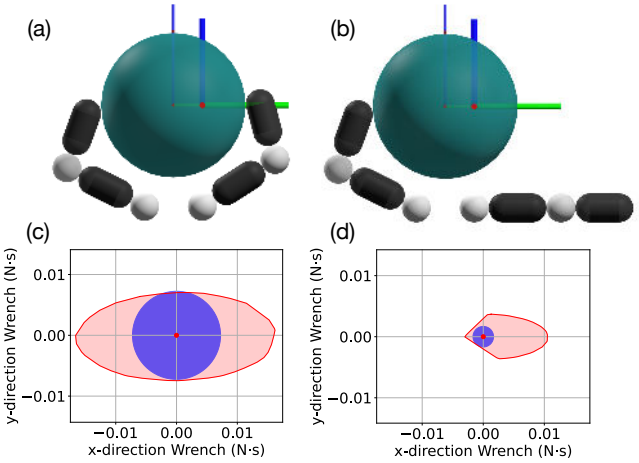


Fig. 15: Top Row: Comparison of two configurations (**a**) and (**b**) for the 2D planar hand system in Figure 2. Although both configurations are advantageous for moving the object to the goal, they share the same MPC value function cost (Table IX). Bottom Row: Plot of the wrench set (illustrated in red) and the maximum inscribed sphere (blue), where (**c**) corresponds to configuration (**a**) and (**d**) corresponds to (**b**). Note that configuration (**a**) has a much larger inscribed sphere due to the anti-podal grasp.

1) A Reduced-Order Model: Inspired by [2, §5.4], instead of solving (38) directly on the full configuration space of q^{a}, we first solve the problem on a reduced-order model, then map the solution to q^{a}. For the Allegro hand, our reduced-order model consists of four spheres, each free to move in 3 dimensions with bounding-box joint limits (Figure 16a). Then, we solve for q^{a} by matching the fingertip positions to those of the spheres with Inverse Kinematics (IK), as shown in Figure 16b. Our IK procedure solves the following Quadratic Program (QP) iteratively,

$$\min_{\delta q, p_{+,k}} \quad \sum_k \|p_{+,k} - p_{\mathrm{des},k}\|^2 \tag{43a}$$

$$\text{s.t.} \quad p_{+,k} = p_k + \frac{\partial p_k}{\partial q}\delta q \tag{43b}$$

$$\mathbf{J}_{n_i}\delta q + \phi_i \geq 0 \tag{43c}$$

$$-\varepsilon \mathbf{1} \leq \delta q \leq \varepsilon \mathbf{1}, \tag{43d}$$

where k indexes each fingertip and its corresponding sphere, $p_{\mathrm{des},k} \in \mathbb{R}^3$ is the location of the sphere, p_k is the location of the fingertips at the current iteration q. Note that (43b) corresponds to a linearization of forward kinematics and (43c) enforces non penetration at every iteration. After an optimal $\delta q^\star$ is found, we start the next iteration with $q \leftarrow q + \delta q^\star$.

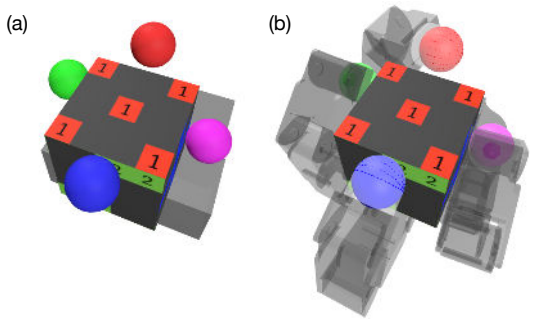


Fig. 16: (**a**) Visualization of the reduced-order system for the Allegro in Section VII-B1, which consists of four spheres. (**b**) Visualization of the IK result on the optimized solution, mapping the reduced-order system to the full system.

2) Contact Sampling Distribution: Furthermore, we noticed that when we sample from the feasible set of (38), the spheres in the reduced-order model frequently do not fully end up in contact with the cube, resulting in a low robustness metric. Thus, after sampling sphere positions from their respective joint limits, we project the spheres to the nearest point on the surface of the cube, so that we are effectively sampling from a distribution of grasps.

C. Results

We test the performance of our method on the AllegroHand system. We first set up three representative pairs of initial and goal configurations for the cube, then run our algorithm to solve (38), whose solution corresponds to the optimal initial configuration of the Allegro hand. Then, we evaluate the fitness of the solution by rolling out MPC (Algorithm 2) from the found initial configuration, and record the error between the final rollout and the goal configurations.

Results in Table X indicate that our method achieves error on the order of 10mm in position, and 30 mrad (1.7°) in orientation. Moreover, the discovered initial hand configurations in Figure 17 agree with our intuition on how the cube should be grasped if we want to move the cube towards the goals.

We note that the given result was obtained with 1000 samples on the reduced-order model. Reducing the number of samples for lower computation time is possible, but would result in higher variance in performance. In addition, we found

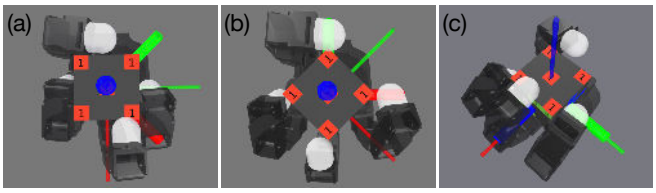


Fig. 17: Initial actuator configurations found by our method for (**a**) yaw 0° to 45°, (**b**) yaw 45° to 90°, and (**c**) pitch 0° to 90°. The initial configuration of the cube is marked with a long and thin triad, while the goal configuration of the cube a short and thick triad.

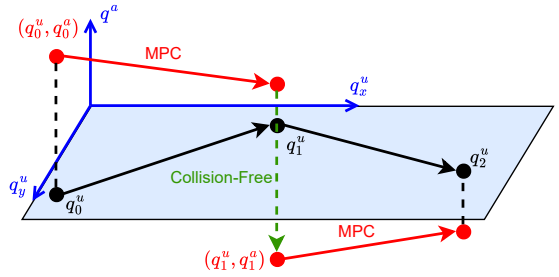


Fig. 18: Illustration of roadmap construction described by Algorithm 4. The vertical axis represents the actuator configuration q^{a}, while the horizontal plane the space of object configurations q^{u}. To connect the vertices $q_0 := (q_0^{\mathrm{u}}, q_0^{\mathrm{a}})$ and $q_1 := (q_1^{\mathrm{u}}, q_1^{\mathrm{a}})$, we first plan from q_0 to q_1^{u} with MPC, reaching the top red dot. As the top red dot and q_1 share the same object configuration q_1^{u}, we can connect them with a collision free planner.

that other types of motions, such as pitch 0° to -90°, or rotation along the x-axis (roll), are physically very difficult on the Allegro hand.

Task	Position [mm]	Rotation [mrad]	Time [s]
Yaw 0° to 45°	12.05 ± 3.49	32.41 ± 27.14	57.83 ± 2.08
Yaw 45° to 90°	9.85 ± 10.88	15.72 ± 21.08	57.28 ± 0.72
Pitch 0° to 90°	21.42 ± 8.91	32.12 ± 32.03	57.84 ± 0.73

TABLE X: Performance of our algorithm for finding initial actuator configurations. Results are obtained with 10 repeated runs. Each run is optimized using 1000 samples.

VIII. Global Planning with Roadmaps

Using the method for generating actuator configurations in Section VII, we present a simple recipe for global search, in which we chain local plans together to efficiently reach *global* goals which are challenging for local MPC. Our method, inspired by the Probabilistic Roadmap (PRM) [73], consists of an offline phase in which the roadmap is constructed, and an online phase where we reach arbitrary goals using the roadmap.

A. Roadmap Construction

In the offline phase, we build a roadmap in which the vertices are grasping configurations and the edges are local plans that transition between these configurations.

We present the roadmap construction method in Algorithm 4. The grasping configurations R in Line 1 are generated by solving (38) for pairs of q^{u}'s from a set of stable object configurations. For each pair of configurations (q_i, q_j) in R, we first try to reach q_j^{u} from q_i by running MPC (Line 5), and then

Algorithm 4: Roadmap Construction

1 **Input:** Grasping configurations $R := \{(q_i^{\mathrm{u}}, q_i^{\mathrm{a}})\}_{i=1}^m$, MPC Parameters: T, $n_{\max}$ and H;
2 $V \leftarrow \emptyset$, $E \leftarrow \emptyset$;
3 **Output:** Sets of vertices V and edges E;
4 **for** *ordered pair* $q_i := (q_i^{\mathrm{u}}, q_i^{\mathrm{a}}), q_j := (q_j^{\mathrm{u}}, q_j^{\mathrm{a}})$ *in* R **do**
5 $\quad q_{0:H}, u_{0:H-1} \leftarrow \mathbf{MPC}(q_i, q_j^{\mathrm{u}}, T, n_{\max}, H)$;
6 $\quad q_{H:M}, u_{H:M-1} \leftarrow \mathbf{CollisionFree}(q_H, q_j)$;
7 $\quad$ **if MPC** *and* **CollisionFree** *both successful* **then**
8 $\quad\quad V$.add(q_i), V.add(q_j);
9 $\quad\quad E$.add$(q_i,\ q_j,\ (q_{0:M}, u_{0:M-1}))$;
10 **return** V, E

reach q_j^{a} by standard collision-free motion planning (Line 6). This procedure is illustrated in Figure 18. We repeat this procedure and add all successful connections to the roadmap.

We further note that in the special case of manipulating objects with geometric symmetries (such as AllegroHand), a single grasping configuration can represent multiple equivalent configurations. For example, a single grasping configuration for a cube can be expanded into 24 distinct configurations, due to the cube's 24 rotational symmetries. By exploiting this property, the same grasp and action sequence can be used to generate multiple edges in the roadmap.

Consider the cube configuration in Figure 19a, which corresponds to the identity rotation (i.e. no rotation). By leveraging symmetry, we can reach its 24 rotational symmetries using just three basic operations: i) yaw from 0° to 90°, ii) yaw from 0° to -90° and iii) pitch from 0° to 90°. These three operations require just five grasps in total: 2 for each yaw and 1 for the pitch (see Figure 17). The grasps take about 5 minutes to generate (Table X). Moreover, connecting each pair of grasping configurations with MPC and collision-free motion planning takes a few seconds (Table VI), keeping the total roadmap construction time under 10 minutes. Parallelizing these steps can further reduce computation time.

To verify the robustness of the constructed roadmap, we conducted a random walk on the roadmap on hardware, and recorded 150 successful consecutive edge transitions before hardware failure occurred (the hand overheated). Recording of this experiment can be found in the supplementary video.

B. Inference on a Roadmap

After the roadmap (V, E) is constructed, we can synthesize plans connecting any starting configuration q_0 to any goal object configuration $q_{\mathrm{goal}}^{\mathrm{u}}$. To do this, we first connect q_0 and $q_{\mathrm{goal}}^{\mathrm{u}}$ to their respective nearest vertices in the vertex set V, which can be done using the same procedure in Line 5 and Line 6 of Algorithm 4. Then, the problem reduces to finding the shortest path between two vertices on a graph, which can be solved with standard methods. An example path generated using this approach is shown in Figure 19. More roadmap planning examples can be found in the supplementary video.

IX. Conclusion

Have we solved the problem of planning and control through contact dynamics? Locally—on CQDC dynamics—the answer

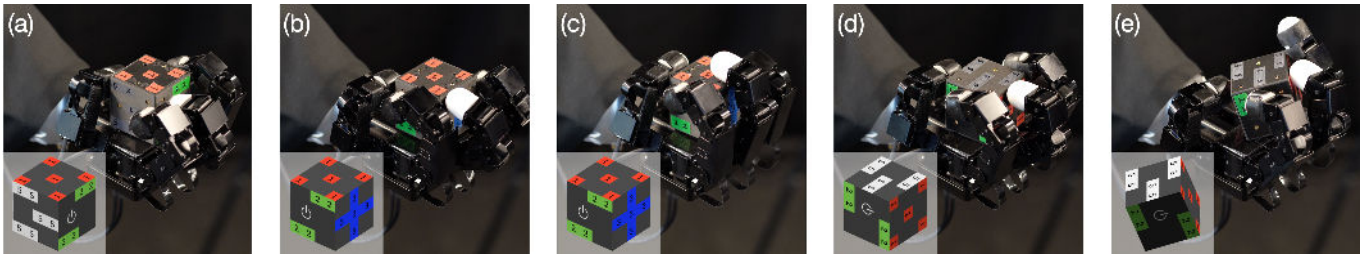


Fig. 19: A complete path generated by our roadmap-based global planner. The object configuration in each frame is highlighted at the lower-left corner. (**a**) shows the starting configuration, which in this example is a vertex already in the roadmap. (**b**) can be reached from (**a**) with a -90° yaw. (**c**) has the same object configuration as (**b**), but the hand has repositioned for a 90° pitch. (**d**) is system configuration after the pitch. (**e**) shows the system reaching the goal configuration from (**d**). The path (**a**)-(**d**) is part of the roadmap and generated offline. (**d**)-(**e**) is generated online using collision-free planning and local MPC.

is tantalizingly close to yes if tracking error is the only thing we care about. However, we do not understand this problem nearly as well as we do the simple pendulum: not only do we lack solid explanations for the small but non-zero number of planner failures, but we also have yet to carefully study the method's control-theoretic properties, including stability and region of attraction.

Much more remains to be understood for second-order dynamics, both in simulation and on hardware. In particular, keeping the robot in contact with the object, without exploiting the artifact of CQDC dynamics, remains one of the biggest unsolved challenges. Incorporating the CTR constraints in MPC greatly alleviates the problem of lost contact, and the initial guess heuristics empirically helps a bit more. However, we occasionally still observe loss of contact, and its root cause remains unclear.

Nevertheless, the tools presented in this paper already enable capabilities that were previously out of reach for model-based methods. By accounting for the unilateral nature of contact, our contact trust region makes it possible to apply a broad range of robotics algorithms to contact-rich manipulation problems. We hope the MPC, grasp synthesis, and roadmap-based global planning methods introduced here are only a small sample of the many contact-rich manipulation algorithms yet to come.

Acknowledgment

We thank Stephen Proulx and Velin Dimitrov for building the cube with active OptiTrack markers, and Giovanni Remigi for making the project website.

Appendix

A. Derivation of (8)

The equality part of the KKT conditions of (6) consists of the stationarity conditon and complementary slackness (6b) and (4c), which can be written as

$$\mathbf{P}q_+ + b - \sum_i \mathbf{J}_i^\top \lambda_i = 0 \tag{44a}$$

$$\nu_i^\top \lambda_i = 0 \quad \forall i. \tag{44b}$$

Looking at the structure of the SOCP parameters $(\mathbf{P}, b, \mathbf{J}_i, c_i)$ tells us that b is the only variable dependent on u. As the primal and dual variables q_+, λ_i are also dependent variables, differentiating both equations w.r.t. u gives us

$$\mathbf{P}\frac{\partial q_+}{\partial u} + \frac{\partial b}{\partial u} - \sum_i \mathbf{J}_i^\top \frac{\partial \lambda_i}{\partial u} = 0 \tag{45a}$$

$$\left(\mathbf{J}_i \frac{\partial q_+}{\partial u}\right)^\top \lambda_i + \frac{\partial \lambda_i}{\partial u}^\top (\mathbf{J}_i q_+ + c_i) = 0 \quad \forall i \tag{45b}$$

Rewriting in matrix form gives us the desired form.

B. Proof of Lemma 2

Writing the wrench set gives us

$$\mathcal{W}(\bar{q}, \bar{u}) = \{w | w = \sum_i \mathbf{J}_{\mathrm{u}_i}^\top \bar{\lambda}_i \tag{46a}$$

$$\bar{\lambda}_i = \mathbf{D}_i \delta u + \lambda(\bar{q}, \bar{u}) \tag{46b}$$

$$\delta u^\top \mathbf{\Sigma} \delta u \leq 1 \tag{46c}$$

$$\bar{\lambda}_i \in \mathcal{K}_i^\star \ \ \forall i \in \mathcal{I}_{\mathrm{uu}} \cup \mathcal{I}_{\mathrm{ue}} \cup \mathcal{I}_{\mathrm{ua}} \tag{46d}$$

$$\mathbf{J}_{\mathrm{u}_i}^\top (q_+^{\mathrm{u}} - \bar{q}^{\mathrm{u}}) \in \mathcal{K}_i \forall i \in \mathcal{I}_{\mathrm{uu}} \cup \mathcal{I}_{\mathrm{ue}} \tag{46e}$$

and we want to prove that the set,

$$\mathcal{M}^{\mathrm{u}}(\bar{q}, \bar{u}) \coloneqq \{q_+ = \mathbf{B}^{\mathrm{u}} \delta u + f(\bar{q}, \bar{u}), \delta u \in \mathcal{F}(\bar{q}, \bar{u})\} \tag{47}$$

is equivalent to

$$\{q_+ | \frac{\epsilon}{h} \mathbf{M}_{\mathrm{u}}(\bar{q})(q_+^{\mathrm{u}} - \bar{q}^{\mathrm{u}}) = h\tau^{\mathrm{u}} + w, w \in \mathcal{W}\}. \tag{48}$$

To prove this, we utilize Lemma 4.3.1 to argue that for any given δu, the next configuration $\bar{q}_+$ and contact impulses $\bar{\lambda}_i$ that are defined by a linear map on δu,

$$\bar{q}_+ = \mathbf{B}^{\mathrm{u}} \delta u + f(\bar{q}, \bar{u}) \tag{49}$$

$$\bar{\lambda}_i = \mathbf{D}_i \delta u + \lambda_i(\bar{q}, \bar{u}), \tag{50}$$

must jointly satisfy the fact that

$$\frac{\varepsilon}{h} \mathbf{M}_u \left(\bar{q}_+^{\mathrm{u}} - \bar{q}^{\mathrm{u}}\right) = h\tau^{\mathrm{u}} + \sum_i \mathbf{J}_{\mathrm{u}_i}^\top \bar{\lambda}_i. \tag{51}$$

References

[1] J. Craig, *Introduction to robotics : mechanics & control*. Reading, Mass.: Addison-Wesley Pub. Co., 1986.

[2] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. USA: CRC Press, Inc., 1994.

[3] M. T. Mason and J. K. Salisbury, *Robot hands and the mechanics of manipulation*. Cambridge, MA, USA: MIT Press, 1985.

[4] D. Prattichizzo and J. C. Trinkle, *Grasping*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 671–700. [Online]. Available: https://doi.org/10.1007/978-3-540-30301-5_29

[5] L. Han, J. Trinkle, and Z. Li, "Grasp analysis as linear matrix inequality problems," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 663–674, 2000.

[6] Y. Ding, A. Pandala, and H.-W. Park, "Real-time model predictive control for versatile dynamic motions in quadrupedal robots," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8484–8490.

[7] 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, 2014.

[8] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. D. Prete, "Optimization-based control for dynamic legged robots," *IEEE Transactions on Robotics*, vol. 40, pp. 43–63, 2024.

[9] H. J. Suh, M. Simchowitz, K. Zhang, and R. Tedrake, "Do differentiable simulators give better policy gradients?" in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 20 668–20 696. [Online]. Available: https://proceedings.mlr.press/v162/suh22b.html

[10] T. Pang and R. Tedrake, "A convex quasistatic time-stepping scheme for rigid multibody systems with contact and friction," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6614–6620.

[11] 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," *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4691–4711, 2023.

[12] T. A. Howell, S. L. Cleac'h, J. Brüdigam, J. Z. Kolter, M. Schwager, and Z. Manchester, "Dojo: A differentiable physics engine for robotics," 2023. [Online]. Available: https://arxiv.org/abs/2203.00806

[13] C. D. Freeman, E. Frey, A. Raichuk, S. Girgin, I. Mordatch, and O. Bachem, "Brax - a differentiable physics engine for large scale rigid body simulation," 2021. [Online]. Available: http://github.com/google/brax

[14] M. Macklin, "Warp: A high-performance python framework for gpu simulation and graphics," https://github.com/nvidia/warp, March 2022, nVIDIA GPU Technology Conference (GTC).

[15] R. Tedrake and the Drake Development Team, "Drake: Model-based design and verification for robotics," 2019. [Online]. Available: https://drake.mit.edu

[16] 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*, 2012, pp. 4906–4913.

[17] T. A. Howell, S. L. Cleac'h, S. Singh, P. Florence, Z. Manchester, and V. Sindhwani, "Trajectory optimization with optimization-based dynamics," 2023. [Online]. Available: https://arxiv.org/abs/2109.04928

[18] 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.

[19] V. Kurtz and H. Lin, "Contact-implicit trajectory optimization with hydroelastic contact and ilqr," 2022. [Online]. Available: https://arxiv.org/abs/2202.13986

[20] N. J. Kong, C. Li, G. Council, and A. M. Johnson, "Hybrid ilqr model predictive control for contact implicit stabilization on legged robots," *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4712–4727, 2023.

[21] D. C. Sorensen, "Newton's method with a model trust region modification," *SIAM Journal on Numerical Analysis*, vol. 19, no. 2, pp. 409–426, 1982. [Online]. Available: https://doi.org/10.1137/0719026

[22] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2006.

[23] J. J. MORE, "Generalizations of the trust region problem," *Optimization Methods and Software*, vol. 2, no. 3-4, pp. 189–209, 1993.

[24] Y. D. Zhong, J. Han, and G. O. Brikis, "Differentiable physics simulations with contacts: Do they have correct gradients w.r.t. position, velocity and control?" 2022. [Online]. Available: https://arxiv.org/abs/2207.05060

[25] K. Werling, D. Omens, J. Lee, I. Exarchos, and C. K. Liu, "Fast and feature-complete differentiable physics for articulated rigid bodies with contact," *arXiv preprint arXiv:2103.16021*, 2021.

[26] A. O. Onol, P. Long, and T. Padır, "Contact-implicit trajectory optimization based on a variable smooth contact model and successive convexification," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2447–2453.

[27] A. Aydinoglu, P. Sieg, V. M. Preciado, and M. Posa, "Stabilization of complementarity systems via contact-aware controllers," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1735–1754, 2022.

[28] 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)*, 2017, pp. 31–38.

[29] 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.

[30] 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.

[31] E. Huang, X. Cheng, and M. T. Mason, "Efficient contact mode enumeration in 3d," June 2020.

[32] A. Aydinoglu and M. Posa, "Real-time multi-contact model predictive control via admm," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 3414–3421.

[33] E. Todorov, "Convex and analytically-invertible dynamics with contacts and constraints: Theory and implementation in mujoco," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 6054–6061.

[34] W. Jin, "Complementarity-free multi-contact modeling and optimization for dexterous manipulation," 2024. [Online]. Available: https://arxiv.org/abs/2408.07855

[35] S. Zhang, W. Jin, and Z. Wang, "Adaptive barrier smoothing for first-order policy gradient with contact dynamics," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 41 219–41 243. [Online]. Available: https://proceedings.mlr.press/v202/zhang23s.html

[36] Y. Shirai, T. Zhao, H. J. T. Suh, H. Zhu, X. Ni, J. Wang, M. Simchowitz, and T. Pang, "Is linear feedback on smoothed dynamics sufficient for stabilizing contact-rich plans?" 2024. [Online]. Available: https://arxiv.org/abs/2411.06542

[37] S. Le Cleac'h, T. A. Howell, S. Yang, C.-Y. Lee, J. Zhang, A. Bishop, M. Schwager, and Z. Manchester, "Fast contact-implicit model predictive control," *IEEE Transactions on Robotics*, vol. 40, pp. 1617–1629, 2024.

[38] H. T. Suh, N. Kuppuswamy, T. Pang, P. Mitiguy, A. Alspach, and R. Tedrake, "Seed: Series elastic end effectors in 6d for visuotactile tool use," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 4684–4691.

[39] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 103, no. 2, pp. 126–133, 06 1981. [Online]. Available: https://doi.org/10.1115/1.3139652

[40] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[41] X. Cheng, E. Huang, Y. Hou, and M. T. Mason, "Contact mode guided motion planning for quasidynamic dexterous manipulation in 3d," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2730–2736.

[42] X. Cheng, S. Patil, Z. Temel, O. Kroemer, and M. T. Mason, "Enhancing dexterity in robotic manipulation via hierarchical contact exploration," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 390–397, 2024.

[43] G. Khandate, T. L. Saidi, S. Shang, E. T. Chang, Y. Liu, S. Dennis, J. Adams, and M. Ciocarlie, "R times r: Rapid exploration for reinforcement learning via sampling-based reset distributions and imitation pre-training," *Autonomous Robots*, vol. 48, no. 7, p. 17, Aug 2024. [Online]. Available: https://doi.org/10.1007/s10514-024-10170-8

[44] OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous in-hand manipulation," 2019. [Online]. Available: https://arxiv.org/abs/1808.00177

[45] T. Chen, J. Xu, and P. Agrawal, "A system for general in-hand object re-orientation," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 297–307. [Online]. Available: https://proceedings.mlr.press/v164/chen22a.html

[46] A. Handa, A. Allshire, V. Makoviychuk, A. Petrenko, R. Singh, J. Liu, D. Makoviichuk, K. Van Wyk, A. Zhurkevich, B. Sundaralingam, Y. Narang, J.-F. Lafleche, D. Fox, and G. State, "Dextreme: Transfer of agile in-hand manipulation from simulation to reality," *arXiv*, 2022.

[47] OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous in-hand manipulation," 2019. [Online]. Available: https://arxiv.org/abs/1808.00177

[48] T. Howell, N. Gileadi, S. Tunyasuvunakool, K. Zakka, T. Erez, and Y. Tassa, "Predictive sampling: Real-time behaviour synthesis with mujoco," 2022. [Online]. Available: https://arxiv.org/abs/2212.00541

[49] A. H. Li, P. Culbertson, V. Kurtz, and A. D. Ames, "Drop: Dexterous reorientation via online planning," *arXiv preprint arXiv:2409.14562*, 2024, available at: https://arxiv.org/abs/2409.14562.

[50] D. E. Stewart and J. C. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with coulomb friction," *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, pp. 162–169 vol.1, 2000. [Online]. Available: https://api.semanticscholar.org/CorpusID:10920958

[51] M. Anitescu, "Optimization-based simulation of nonsmooth rigid multibody dynamics," *Mathematical Programming*, vol. 105, no. 1, pp. 113–143, Jan 2006. [Online]. Available: https://doi.org/10.1007/s10107-005-0590-7

[52] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.

[53] X. Han, J. Masterjohn, and A. Castro, "A convex formulation of frictional contact between rigid and deformable bodies," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6219–6226, 2023.

[54] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[55] M. Mason, *Mechanics of Robotic Manipulation*. MIT Press, 2001.

[56] Y. Shirai, T. Zhao, H. T. Suh, H. Zhu, X. Ni, J. Wang, M. Simchowitz, and T. Pang, "Is linear feedback on smoothed dynamics sufficient for stabilizing contact-rich plans?" 2024.

[57] J. J. Moré and D. C. Sorensen, "Computing a trust region step," *SIAM Journal on Scientific and Statistical Computing*, vol. 4, no. 3, pp. 553–572, 1983. [Online]. Available: https://doi.org/10.1137/0904038

[58] L. Niu and Y. Yuan, "A new trust-region algorithm for nonlinear constrained optimization," *Journal of Computational Mathematics*, vol. 28, no. 1, pp. 72–86, 2010. [Online]. Available: http://global-sci.org/intro/article_detail/jcm/8508.html

[59] M. T. Mason, "Mechanics and planning of manipulator pushing operations," *Int. J. Rob. Res.*, vol. 5, no. 3, p. 53–71, Sep. 1986. [Online]. Available: https://doi.org/10.1177/027836498600500303

[60] 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.

[61] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*, 1992, pp. 2290–2295 vol.3.

[62] K. Lynch and F. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge university press, 2017.

[63] M. Erdmann, "On a representation of friction in configuration space," *The International Journal of Robotics Research*, vol. 13, no. 3, pp. 240–271, 1994. [Online]. Available: https://doi.org/10.1177/027836499401300306

[64] H. Dai, A. Majumdar, and R. Tedrake, *Synthesis and Optimization of Force Closure Grasps via Sequential Semidefinite Programming*. Cham: Springer International Publishing, 2018, pp. 285–305. [Online]. Available: https://doi.org/10.1007/978-3-319-51532-8_18

[65] T. Pang and R. Tedrake, "Easing reliance on collision-free planning with contact-aware control," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8375–8381.

[66] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems," in *First International Conference on Informatics in Control, Automation and Robotics*, vol. 2. SciTePress, 2004, pp. 222–229.

[67] M. H. Raibert and J. J. Craig, "Hybrid Position/Force Control of Manipulators," *Journal of Dynamic Systems, Measurement, and Control*, vol. 103, no. 2, pp. 126–133, 06 1981. [Online]. Available: https://doi.org/10.1115/1.3139652

[68] M. ApS, *MOSEK Optimizer API for Python 10.2.1*, 2024. [Online]. Available: https://docs.mosek.com/latest/pythonapi/index.html

[69] R. Tedrake, *Underactuated Robotics*, 2023. [Online]. Available: https://underactuated.csail.mit.edu

[70] A. Castro, X. Han, and J. Masterjohn, "A theory of irrotational contact fields," *arXiv preprint arXiv:2312.03908*, 2023.

[71] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, 2024.

[72] A. H. Li, P. Culbertson, J. W. Burdick, and A. D. Ames, "Frogger: Fast robust grasp generation via the min-weight metric," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 6809–6816.

[73] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, and W. Burgard, *Principles of robot motion: theory, algorithms, and implementations*. MIT press, 2005.