

Approximating Global Contact-Implicit MPC via Sampling and Local Complementarity

Sharanya Venkatesh*, Bibit Bianchini*[†], Alp Aydinoglu, William Yang, Michael Posa

*The first two authors contributed equally to this work. [†]Corresponding author (bibit@seas.upenn.edu).

Abstract—To achieve general-purpose dexterous manipulation, robots must rapidly devise and execute contact-rich behaviors. Existing model-based controllers cannot globally optimize in real time over the exponential number of possible contact sequences. Instead, progress in contact-implicit control leverages simpler models that, while still hybrid, make local approximations. Locality limits the controller to exploit only nearby interactions, requiring intervention to richly explore contacts more broadly. Our approach leverages the strengths of local complementarity-based control combined with low-dimensional, but global, sampling of possible end effector locations. Our key insight is to consider a contact-free stage preceding a contact-rich stage at every control loop. Our algorithm, in parallel, samples end effector locations to which the contact-free stage can move the robot, then considers the cost predicted by contact-rich MPC local to each sampled location. The result is a globally-informed, contact-implicit controller capable of real-time dexterous manipulation. We demonstrate our controller on precise, non-prehensile manipulation of non-convex objects with a Franka arm. Project website: <https://approximating-global-ci-mpc.github.io>

Index Terms—Optimization and optimal control, contact modeling, contact-implicit model predictive control.

I. INTRODUCTION

FOR multi-purpose robots to successfully deploy into the home and workplace, they will need to be capable of dexterous manipulation of complex objects. Progress towards this goal has been made on many fronts, including imitation learning from human demonstrations [1] and offline model-based planning [2], [3]. However, these approaches both require advanced knowledge of the task and can fail to generalize to even minor permutations, such as new goal configurations.

An alternative approach that has made recent strides is contact-implicit model predictive control (CI-MPC). Contact-implicit methods optimize for state and/or input trajectories and a corresponding sequence of contact modes. Facing hybrid, nonlinear dynamics, CI-MPC approaches compromise to reach real-time rates, such as by picking one of many sampled trajectories [4], [5], or using simplified dynamics. One such simplified model that captures the hybrid aspect of contact-rich dynamics is a linear complementarity system (LCS) [6], a

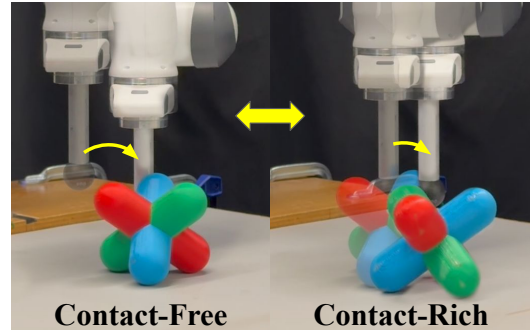


Fig. 1. Our real-time CI-MPC combines a global, explorative contact-free stage with a local contact-rich stage. At every control loop, our algorithm chooses contact-rich actions to make goal progress, or contact-free actions to pursue more amenable starting locations for future contact-rich actions.

piecewise-linear representation. However like many simplified models, LCS dynamics are limited in accuracy to a local neighborhood of the true underlying system (Fig. 3).

Our novel CI-MPC approach inherits local efficacy of complementarity-based control and global efficacy of sampling-based strategies, only requiring sampling individual robot collision geometry locations. This can be a tractably low-dimensional space, e.g. 3D end effector locations. Our key insight is that all contact-rich control can be split into a contact-free stage when the robot moves in a collision-free path (easier to compute than contact-rich trajectories), followed by a contact-rich stage when the robot makes and breaks contact freely (Fig. 1). Existing CI-MPC methods can be effective in the contact-rich stage, if the local neighborhood is amenable to making progress. Sampling is a natural solution: we sample end effector locations where the contact-free-to-rich transition can occur, quantifying sample advantages by their local CI-MPC optimization costs. In closed loop with cost- and progress-based switching logic, our controller autonomously switches between contact-free and contact-rich modes, trading off future investment with immediate progress, respectively.

In this paper, we contribute:

- An online, multi-contact manipulation method combining global sampling with local control, requiring no offline computation or training. Our approach accomplishes tasks that local control alone essentially always fails to do.
- Hardware results of our algorithm achieving pose goals of non-convex objects to high precision with a Franka arm, using 1 robot-object and 3 object-environment contacts.
- Ablations and comparisons in simulation, including to MuJoCo MPC (MJPC) [5], another non-local CI-MPC method our approach outperforms in success and safety.

Manuscript received: May 13, 2025; Revised August 4, 2025; Accepted September 5, 2025. This paper was recommended for publication by Editor Júlia Borràs Sol upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by a National Defense Science and Engineering Graduate (NDSEG) Fellowship, an NSF CAREER Award (FRR-2238480), and the RAI Institute.

S. Venkatesh, B. Bianchini, and M. Posa are with the GRASP Laboratory at the University of Pennsylvania, Philadelphia, PA 19104.

A. Aydinoglu is with Boston Dynamics, Waltham, MA 02451.

W. Yang is with Amazon Robotics, Seattle, WA 98109.

Digital Object Identifier (DOI): see top of this page.

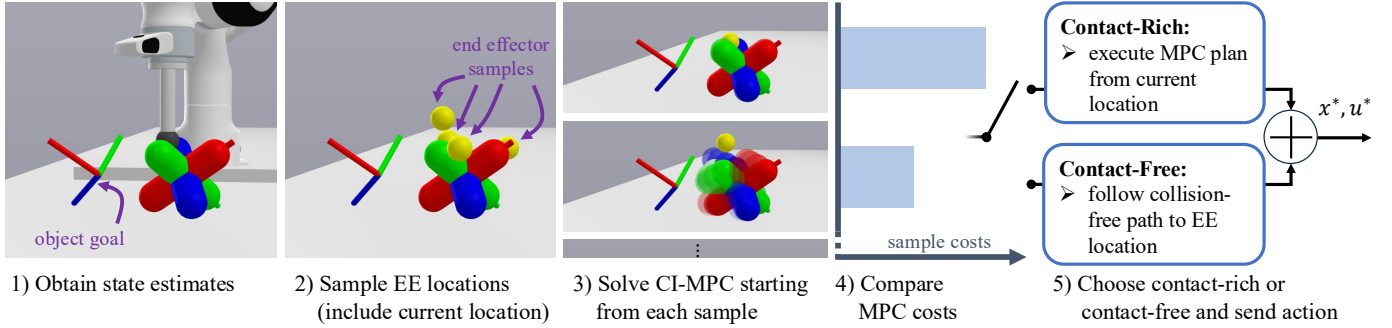


Fig. 2. The algorithm for one control loop of our sampling-based contact-implicit controller. The third step that solves a local CI-MPC problem for each sample can be parallelized, since each plan is independent. In this example, the top sample’s CI-MPC plan makes little progress to the object goal, and thus is associated with high cost. The second sample’s CI-MPC plan shows more progress, and thus is lower cost.

II. RELATED WORK

A. Offline Trajectory Optimization

Solving a global nonlinear control problem can be too time-intensive to compute online, but offline contact-implicit trajectory optimization can be useful [7], [8]. Non-smooth contact dynamics can create confront challenging cost landscapes [9]. Mitigations include artificially smoothing gradients [3], [10], graph search [2] or sampling [11] over contact mode sequences and/or control inputs [12], or restricting applications to simpler 2D problems [13]. Task and motion planning (TAMP) (e.g. [14]) is a flavor of making high-level choices combined with lower-level trajectory optimization. There is a long history of combining global sampling and local optimization (in robotics, see [9], [15], among others). Despite some applications to robotic manipulation [16], to the best of our knowledge these methods have not achieved real-time contact-rich control.

B. Contact-Implicit MPC

Recent progress has reformulated global tasks into more tractable online MPC problems. One approach uses local complementarity-based approximations to the global system [17], [18], encouraging consensus between local non-smooth rigid-body contact constraints and local dynamics, while exploring contact modes via small-scale mixed integer optimization. Other distinct but similar approaches use artificially smoothed contact dynamics, allowing online gradient-based algorithms to explore different contact modes [19]–[22]. All of these methods, however, leverage local approximations and thus may fundamentally struggle to escape local minima and initiate distant, but beneficial, contact without intervention.

C. The Role of Sampling for Planning and Control

While the non-smooth dynamics of contact-rich scenarios pose serious challenges to gradient-based techniques, gradient-free methods like random sampling have shown promise. Many methods in the literature include sampling in the spaces of input trajectories [5], [23], contact mode sequences [11], [24], or both [12]. In our work, we explore a lower-dimensional form of sampling: sampling end effector configurations, using these as global seeds for the local contact-implicit controller from [18]. This does not require sampling full trajectories nor full system states, but retains the primary benefit of sampling as a way out of local minima.

III. BACKGROUND

We review modeling choices for contact dynamics (§III-A), introduce a contact-rich optimal control problem (§III-B), then detail C3 [18], an existing real-time, contact-implicit algorithm that locally approximates the control problem (§III-C).

A. Contact Dynamics

For a contact-rich system with state x and inputs u subject to contact forces λ , its dynamics can be generally written as

$$x_{k+1} = g(x_k, u_k, \lambda_k), \quad (1a)$$

$$0 \leq \lambda_k \perp \pi(x_k, u_k, \lambda_k) \geq 0, \quad (1b)$$

where the discrete-time dynamics g depend on the contact forces λ_k , which are the solution to a nonlinear complementarity problem (NCP) [25] in (1b). The NCP elegantly embeds the multi-modal nature of contact-rich systems.

In our context, the vector λ_k represents contact forces, and the complementarity constraint (1b) enforces the hybrid (or non-smooth) aspects of the dynamics; for example, it can encode the constraint that forces must only occur when in contact, and can enforce the stick-slip effects of Coulomb friction. For simulation, the solved contact forces are found to satisfy contact dynamics encoded by the NCP in (1b) and affect the system dynamics as in (1a).

A local approximation to (1) that still preserves the multi-modality of contact is a linear complementarity system (LCS) [6]. An LCS describes the state and contact force trajectories for an input sequence starting from x_0 such that

$$x_{k+1} = Ax_k + Bu_k + D\lambda_k + d, \quad (2a)$$

$$0 \leq \lambda_k \perp Ex_k + F\lambda_k + Hu_k + c \geq 0. \quad (2b)$$

where $x_k \in \mathbb{R}^{n_x}$, $\lambda_k \in \mathbb{R}^{n_\lambda}$, $u_k \in \mathbb{R}^{n_u}$, $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $D \in \mathbb{R}^{n_x \times n_\lambda}$, $d \in \mathbb{R}^{n_x}$, $E \in \mathbb{R}^{n_\lambda \times n_x}$, $F \in \mathbb{R}^{n_\lambda \times n_\lambda}$, $H \in \mathbb{R}^{n_\lambda \times n_u}$, and $c \in \mathbb{R}^{n_\lambda}$.

In computing the LCS representation of a given system, we consider a nominal x_{nom} , u_{nom} and the resulting λ_{nom} from solving (1b). The LCS dynamics (2) approximate the non-linear hybrid dynamics (1) by linearizing g and π .

Once the LCS is constructed, for a given x_k and u_k , the corresponding complementarity variable λ_k is found by solving (2b). Similarly, x_{k+1} can be computed using (2a) when x_k , u_k , and λ_k are known. Given an input u_k and state x_k , the next state is $x_{k+1} = \text{LCS}(x_k, u_k)$ defined by (2).

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

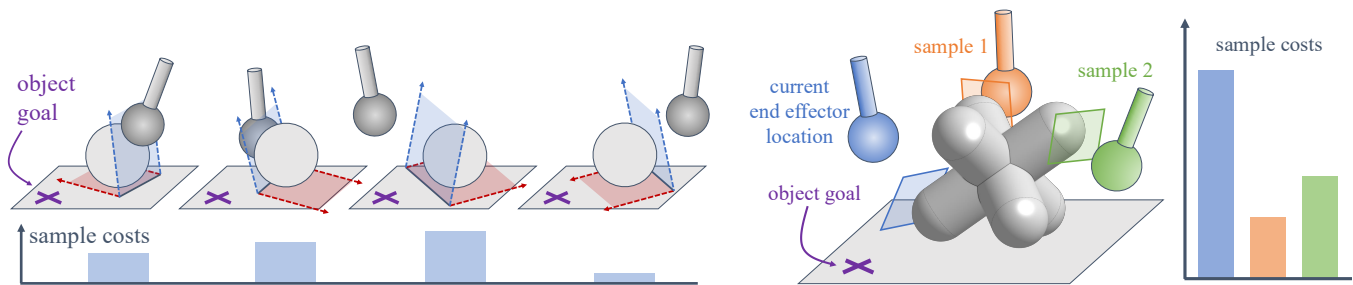


Fig. 3. Left: A spherical end effector approaches a spherical object on a flat table. Loosely speaking, the LCS approximates object geometry as a set of hyperplanes coincident with and tangent to their witness points with respect to other geometries of interest. The hyperplane for ground-object contact is in red, while the robot-object contact hyperplane is in blue. Each initial condition has its associated MPC cost. In this example, the rightmost sample’s LCS approximation allows the robot to most effectively foresee progressing the object toward the goal and correspondingly has the lowest sample cost. Right: These LCS approximations are well-defined even for more complicated geometries, only requiring witness points between the object and other collision geometry.

B. Optimal Control through Contact

We are interested in the MPC problem first posed in [8] for contact-rich systems with dynamics described by an NCP,

$$\min_{x_k, u_k, \lambda_k} \sum_{k=0}^{N-1} (x_k^T Q_k x_k + u_k^T R_k u_k) + x_N^T Q_N x_N \quad (3)$$

$$\text{s.t. } x_{k+1} = g(x_k, u_k, \lambda_k), \quad (4)$$

$$0 \leq \lambda_k \perp \pi(x_k, u_k, \lambda_k) \geq 0, \quad (5)$$

$$x_0 = x_{\text{init}}, (x_k, u_k) \in \mathcal{C}. \quad (6)$$

This optimization problem solves for state $x_{0:N}$, control input $u_{0:N-1}$, and contact force $\lambda_{0:N-1}$ trajectories that satisfy the system’s dynamics (4), contact constraints (5), and initial/other constraints (6), e.g. input limits or safety constraints.

Solving this MPC problem for even simple multi-contact systems is intractable at real-time rates due to the complicated nature of the discontinuous contact dynamics (5) and their impact on system dynamics (4). An alternative is to approximate them with expressions that are faster to evaluate.

C. Consensus Complementarity Control (C3)

To simplify (3)-(6), we pose the control problem with the dynamics expressed as an LCS (substituting (2) for (4) and (5)) using x_{init} (and some nominal u_0, λ_0) as the LCS linearization point. This is still computationally intensive because of the complementarity constraint (2b). Consensus Complementarity Control (C3) [18], without guarantees of convergence, can approximate this optimization problem quickly. C3 uses the alternating direction method of multipliers (ADMM) to iteratively solve the problem while satisfying only the dynamics (2a) then only the contact constraints (2b). This first stage is a quadratic program (QP) with linear equality and inequality constraints. The second stage can be parallelized across time steps, since the contact constraints (2b) depend only on step k . Thus, the second stage can be formulated as a small-scale, sufficiently fast mixed-integer QP (MIQP). With extra cost terms to encourage consensus between these two stages, C3 can converge to the solution to the optimization problem. In practice, using a suboptimal solution at a faster rate is more performant than a more optimal solution at a slower rate, so it is beneficial to terminate C3 after a few ADMM iterations.

IV. METHODS

We extend the capabilities of local control by injecting global insights into the problem. Using an LCS model as in C3 [18] has negative consequences that motivate our approach (§IV-A). The key is to split the problem into an initial contact-free mode and a subsequent contact-rich mode (§IV-B). To approximate the new bilevel optimization problem in real time, we sample low-dimensional end effector locations at the mode switch (§IV-C). In closed-loop, our controller autonomously switches between modes based on cost and progress (§IV-D). While our motivation and demonstrations use C3 [18] as the local contact-rich controller, our hierarchical approach could wrap around other contact-rich controllers, e.g. [20].

A. Limitations of LCS as an MPC Modeling Choice

The LCS approximation of a system (as in [18], [20]) creates hyperplanes in (x, u, λ) space with respect to every contact pair of interest. While not strictly accurate¹, one can conceptualize these hyperplanes as approximating the non-linear geometry of an object via witness point tangent planes in \mathbb{R}^3 (Fig. 3). With these visuals, it is evident the half-space boundaries can easily lock the robot into regions where it cannot apply positive normal force to move the object towards its goal (e.g. third from left in Fig. 3). Friction cone constraints prevent planning negative normal forces, so the robot will plan to do nothing.

Thus, any LCS- or linearization-based controller is fundamentally local. Past C3 demonstrations [17], [18] required careful experimental design and/or additional heuristics to encourage or force the end effector out of regions where the local LCS view is antagonistically simplified. These heuristics depend on the system and the goal, and bake in prior knowledge for what regions better enable task progress. We seek a simple way to instill global knowledge into a controller that is locally effective.

B. Separate Contact-Free and Contact-Rich Modes

First, consider that any optimal multi-contact control problem can be decomposed into an initial contact-free mode and

¹Precisely, the linearization of the gap function ϕ is a combination of the hyperplanes in the figure, with additional terms derived from the linearizations of the single-step dynamics.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

a subsequent contact-rich mode. The contact-free mode has no collisions between the robot and manipulated objects, and the contact-rich mode can make and break contact as it pleases. This suggests a new bilevel optimization problem² given by

$$\min_{K, x_{\text{switch}}} \left(\min_{x_k, u_k} \sum_{k=0}^K c_{\text{free}}(x_k, u_k) + \min_{x_k, u_k} \sum_{k=K}^N c_{\text{rich}}(x_k, u_k) \right), \quad \text{s.t. } x_K = x_{\text{switch}} \quad \text{s.t. } x_K = x_{\text{switch}} \quad (7)$$

whose outer problem selects a time step K and system state x_{switch} for *when and where* the mode switches from contact-free to contact-rich, and whose inner problem minimizes the contact-free and contact-rich costs (c_{free} and c_{rich} , respectively) given the intermediate state x_{switch} . This loses no generality from the original contact-rich control problem (3)-(6). To enable computational efficiency, we approximate in (7):

- 1) the contact-free mode as end effector repositioning (kinematic path planning);
- 2) the contact-rich mode via local CI-MPC (C3 [18]).

The first approximation can be justified for quasi-static systems, where the object remains stationary during contact-free motion. While no tasks are truly quasi-static, this is a common assumption in manipulation [26]–[28]. As we will see in §V, even for more dynamic tasks, this is not particularly limiting. This means the effective output of the first stage is a new configuration for the robot collision geometry. This configuration space can be tractable for optimization or search if we use a low-dimensional representation, such as a 3D end effector location (as used in our demonstrations). Specifically, if a full system state x contains $[q_{\text{ee}}, q_{\text{obj}}, v]$, we impose $x_{\text{switch}} = [q_{\text{ee,switch}}, q_{\text{obj,init}}, v_{\text{init}}]$ where only $q_{\text{ee,switch}}$ differs from x_{init} . Additionally, the cost of the contact-free stage can be penalized by a simple end effector travel cost.

The second simplification is based on the efficacy of C3 for the more challenging portion of the problem, with LCS dynamics (including hyperplane definitions, as in Fig. 3) local to the starting state with x_{switch} . By making global decisions via contact-free path planning and sequencing with contact-rich motions, we are able to mitigate the limitations inherent in the locality of C3’s contact-implicit control.

C. Sample Switching States

Even with simple travel cost and C3 approximations to the contact-free and contact-rich modes, respectively, the bilevel optimization problem is intractable to solve in real time because it would require solving C3 for every candidate x_{switch} . Instead, we sample a small set of x_{switch} , which crucially can be evaluated in parallel. This sample set always includes the current system state, i.e. our hierarchical controller always knows the contact-rich plan and can enter/remain in the contact-rich mode whenever it wants. If the controller is currently in contact-free mode, the sample set additionally includes the actively pursued $q_{\text{ee,switch}}$ (called $q_{\text{ee,switch,prev}}$ coming from the previous loop’s x_{switch} i.e. $x_{\text{switch,prev}}$) to continually update its sample cost. At every control loop, our controller computes

the C3 costs for the x_{switch} candidates, adding the contact-free travel costs appropriately. We assume access to a sampling strategy $\text{SampleEE}(x_{\text{init}}, x_{\text{switch,prev}}, n_s)$ which draws n_s x_{switch} samples for potential end effector locations. The first sample is always $q_{\text{ee,init}}$, and the second is $q_{\text{ee,switch,prev}}$, if there is one. To evaluate costs, we assume access to the C3 algorithm and the function $\text{C3Cost}(x_{\text{sample}}, x_{\text{goal}})$. This sampling and evaluation step is detailed by Algorithm 1.

Algorithm 1 Sample and Evaluate

Require: $x_{\text{init}}, x_{\text{goal}}, x_{\text{switch,prev}}$ if there was one, number of samples n_s , sampling strategy SampleEE , travel cost weight w_{travel} , C3 algorithm and associated cost C3Cost

- 1: $q_{\text{sample,ee},1:n_s} \leftarrow \text{SampleEE}(x_{\text{init}}, x_{\text{switch,prev}}, n_s)$

Parallelizing samples, solve C3 and impose travel cost.

- 2: **for** $i \in \{1, \dots, n_s\}$ **do**
 - 3: $c_{\text{travel}} \leftarrow w_{\text{travel}} \|q_{\text{ee,init}} - q_{\text{ee,sample},i}\|$
 - 4: $x_{\text{sample},i} \leftarrow [q_{\text{ee,sample},i}, q_{\text{obj,init}}, v_{\text{init}}]$
 - 5: $c_{\text{sample},i} \leftarrow \text{C3Cost}(x_{\text{sample},i}, x_{\text{goal}}) + c_{\text{travel}}$
 - 6: **end for**
 - 7: **return** $(x_{\text{sample}}, c_{\text{sample}})_{1:n_s}$
-

D. Switch Modes Based on Costs and Progress

Given a set of x_{switch} candidates and associated inner costs c_{sample} , we must pursue one. If we pick the current state, our controller executes the actions computed during the C3 solve. If we pick a different state, our controller commands a collision-free path to the new end effector location. A natural selection algorithm might be to choose the sample with the lowest associated cost. However, we find this results in indecisive behavior. We attribute this primarily to C3 variability, as C3 is a local algorithm with no convergence guarantees. To mitigate these effects, we add hysteresis on cost comparisons to keep the controller in the current mode, barring a clearly better alternative. We define hysteresis values $h_{\text{free-to-rich}}$, $h_{\text{rich-to-free}}$, and $h_{\text{free-to-free}}$ (which relates to switching between different relocation targets within the contact-free mode). We choose $h_{\text{rich-to-free}}$ to be relatively large, so that the algorithm will not mistakenly abandon a contact-rich mode that is progressing toward the goal. However, this alone creates the risk of the controller becoming stuck while no progress is made. To address this, we force a transition to the contact-free mode if the manipulated object fails over a specified period of time to make progress toward the goal. When this occurs, the controller pursues the sampled x_{sample} with the lowest cost.

To increase the likelihood of pursuing a high-quality x_{sample} after leaving contact-rich mode, we maintain a buffer of relevant sampled end effector positions and their associated costs. The sample buffer is pruned of samples whose associated object locations at the time of computing cost are too far from the current object location. At every control loop, we:

- 1) Update x_{init} from proprioception/vision-based sensing.
- 2) Generate and evaluate samples via Algorithm 1.
- 3) Maintain the sample buffer: remove outdated samples based on object movement and add those from Step 2.

²Ignoring other constraints for brevity, including dynamics, contact dynamics, initial condition, input limits, etc.

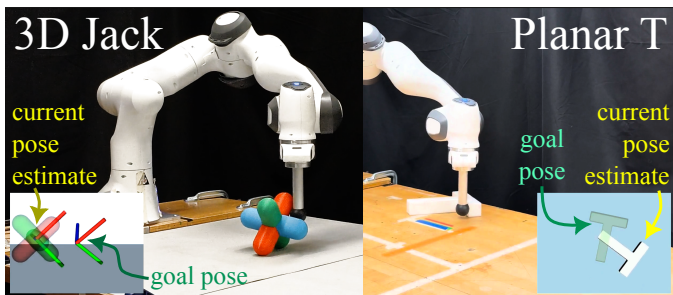


Fig. 4. We demonstrate our controller on two manipulation examples: 3D pose goals with a jack (left), and 2D planar pose goals with a T (right). The inset renderings in the bottom corners depict a view of the object’s goal pose relative to the current pose estimate.

- 4) Determine the controller mode. Select the lowest cost sample from the buffer, accounting for hysteresis which privileges the current mode. If no progress has been made, force a switch to contact-free mode.
- 5) Execute a plan based on the current mode. If in contact-rich mode, execute the plan from solving C3 with x_{init} . If in contact-free mode, follow a collision-free path from x_{init} to the pursued sample.

We repeat this with receding horizon, enabling the control to adapt to observed dynamics, considering more samples with every control loop. Fig. 2 depicts one control loop.

V. EXPERIMENTS

A. Task, State Representation, and Contact Modeling

To validate our controller, we test with a Franka Panda arm equipped with a spherical end effector to manipulate an object to a pose goal. When the object goal is reached within position and rotation tolerances, a new pose goal is randomly generated, demonstrating generalization over initial and goal poses. We test on two hardware examples (3D jack and planar T, both shown in Fig. 4) and on our 3D jack example in simulation for more direct comparison to baselines (more detail in §V-E). Our controller operates on LCS dynamics where $x \in \mathbb{R}^{19}$ (end effector position, object position, object orientation, and their velocities), $u \in \mathbb{R}^3$ (cartesian forces applied to the end effector), and $\lambda \in \mathbb{R}^{16}$ (4 contact pairs with a 4-sided polyhedral friction cone approximation [29]).

Our approach requires manual enumeration of contact *geometries*, but uses collision detection between them to identify possible contact *points* at every control loop. These points automatically change with motion, and the approach can accommodate any geometries handled by the collision detection algorithm. For both examples, one contact pair uses the closest witness point on the object to the end effector, and the remaining 3 contact pairs are for object-ground contacts. For the jack, we obtain the closest point per capsule to the ground. A visualization of all the jack contact points for a particular configuration is illustrated in Fig. 5. For the T, we define small spheres at the 3 distal ends as witnesses to the ground. For this planar example, we use the full 3D states of the end effector and T but add a constraint in the contact-rich stage (C3) to enforce the end effector maintains a pre-defined height.

B. Implementation

Our controller and accompanying diagram (Fig. 6) is implemented in C++ in Drake [30]. As in [17], an operational space controller (OSC) [31] tracks task-space commands specified by our policy at 8-12Hz, via joint-level control at 1kHz. FoundationPose [32] estimates object state and runs at 30Hz with a D455 RealSense RGBD camera. Our setup uses two computers: a 13th generation Intel Core i9-13900KF with 32 threads (for our controller) and an NVIDIA GeForce RTX 4090 GPU (for FoundationPose), and an Intel i7-8700K processor (for our OSC and robot drivers) equipped with a real-time kernel for communicating with the Franka. Inter-computer communication occurs over LCM [33].

C. Sampling Parameters

We consider three samples (including the end effector’s current location) in parallel with every control loop ($n_s = 3$). We solve C3’s QP with OSQP [34] and MIQP with Gurobi [35]. Our samples and the C3 MIQP both utilize parallelization, so we use the maximum number of threads in our hardware setup. Because of this, additional samples slow down the control rate and decrease performance. With this setup, our controller runs at 8-12Hz for all experiments. We randomly sample end effector locations on a sphere around the jack and on an enlarged planar perimeter for the T. While more sophisticated sampling is compatible with our approach, these simple distributions are notably effective.

D. Cost Parameters

To avoid aggressive behavior from high costs, we truncate the final goal’s displacement from the current object location (15cm, 2 radians). This bounds the errors encountered by the MPC and mitigates indecision when orientation error approaches π radians, at which any rotation direction is equally effective. This orientation truncation requires hysteresis on the rotation axis to ensure stability of the direction of rotation.

For the Q cost matrix, a typical diagonal structure is sufficient for position and velocity errors in our experiments. However, orientation presents some challenges. The true orientation error we desire to minimize is θ_{error}^2 , where θ_{error} is the scalar angle of the relative rotation between the current and goal orientations. From quaternions, this is calculated as

$$\theta_{error} = \left(\arctan \left(\frac{\|q_{rel,x}^2 + q_{rel,y}^2 + q_{rel,z}^2\|}{q_{rel,w}} \right) \right)^2, \quad (8a)$$

$$\text{where } q_{rel} = q_{quat,curr}^{-1} \otimes q_{quat,goal}, \quad (8b)$$

for \otimes as quaternion product. The arctan indicates a problematic region where its argument is zero – this occurs precisely when $q_{quat,curr} = q_{quat,goal}$. The landscape is not strictly convex at this point and is locally non-convex. Thus, the naive 2-norm error between the elements of $q_{quat,curr}$ and $q_{quat,goal}$,

$$\tilde{\theta}_{error}^2 = \|q_{quat,curr} - q_{quat,goal}\|^2. \quad (9)$$

poorly captures the true θ_{error} when it is small. To address this, we set the 4x4 object quaternion portion of Q (throughout

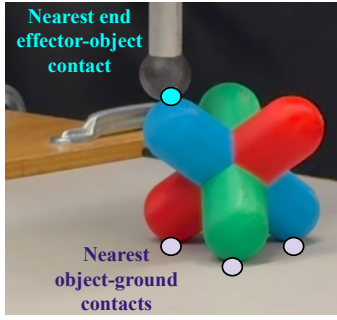


Fig. 5. Illustration of contact points considered for the jack object.

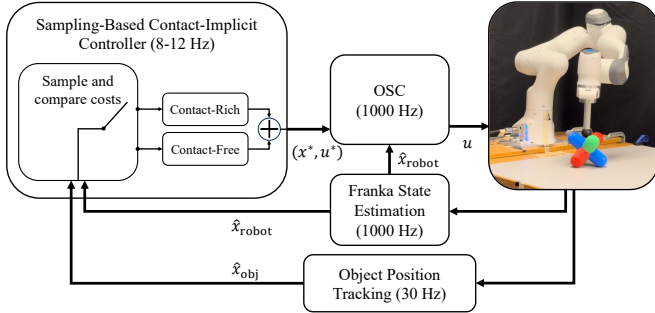


Fig. 6. A block diagram depicts our hierarchical controller (top left block), which performs real-time CI-MPC globally by sampling potential end effector locations for switching from contact-free to contact-rich mode.

the entire MPC horizon) to be the Hessian of θ_{error}^2 with respect to the elements of the current quaternion, about the $q_{\text{quat,curr}}, q_{\text{quat,goal}}$ operating point. We regularize this Hessian, adding $|\gamma| \cdot \mathbb{I}_{4 \times 4}$, where γ is its most negative eigenvalue to ensure positive-semi definiteness. Implementing this portion of Q is a critical step to effectively and reliably track orientation.

E. Comparisons

Due to the inability to escape geometric local minima, C3 fails essentially 100% of the time on our tasks and thus is not compared. We compare with two ablations. Naive-R randomly selects an x_{sample} when the controller is unproductive; in contrast, our controller selects the x_{sample} with the lowest associated CI-MPC cost and can switch modes due to progress or cost. Naive-B only considers the x_{sample} behind the object from the goal, testing whether a trivial region of space around the object is sufficient or if broader exploration is required.

Additionally, we compare with MuJoCo MPC (MJPC) with predictive sampling [5] on the 3D jack task in simulation. As with our controller, we use MJPC as an online planner operating on a reduced model, abstracting the end effector as controlled in xyz only, whose motion is tracked via our joint-level OSC simulated in Drake. Unlike our controller, the MJPC planner models the nonlinear dynamics of the floating end effector moving in 3D with the jack and environment. Any comparison is sensitive to tuning, and we put forth a best-faith effort to tune MJPC. We found tighter control input limits to be helpful in preventing wild motions, selected a noise parameter of 0.295 to balance sample exploration and previous plan exploitation, and tuned state and input costs. MJPC performed best with control input splines of 5 knot

3D Jack Hardware Goals Achieved within Time Limit

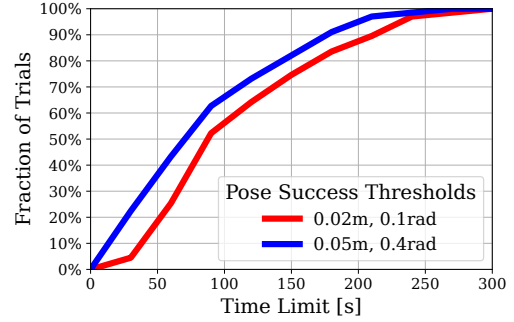
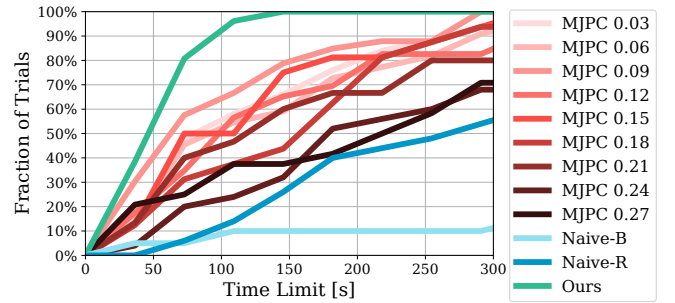


Fig. 7. Cumulative distribution for time to goal, using sets of tight and loose position and orientation tolerances.

3D Jack Simulation, Goal Fraction Achieved Within Time Limit



3D Jack Simulation, Goal Outcomes

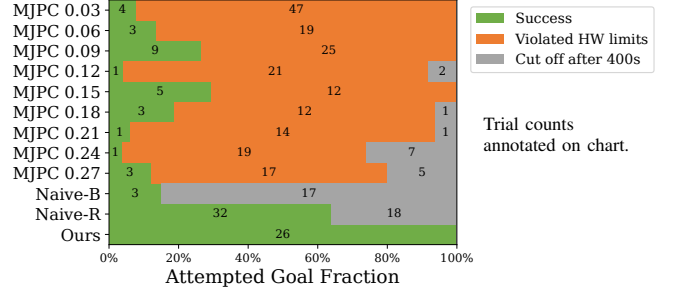


Fig. 8. Our approach outperforms all other baselines in 3D jack manipulation in simulation with high precision tolerances (2cm and 0.1 radians pose error). Our controller achieves goals faster than MJPC (top) while also avoiding hardware limit violations (bottom) such as joint velocity, joint torque, and workspace limits. The MJPC lines are annotated with the values of the end effector velocity cost weight, showing decreased performance but ineffective ability to reduce hardware violations at higher end effector velocity costs. Ablations of our approach (Naive-R and Naive-B) avoid hardware limit violations (bottom) but are substantially less performant in terms of success rate and time-to-goal (top).

points over a predictive horizon of 0.8 seconds, reasonably longer than our policy's 0.25 seconds, since we effectively get longer-term insight via our contact-free sampling.

VI. RESULTS

See our project webpage for extended results. We evaluate performance under two sets of success thresholds: the tight set requires under 2cm and 0.1 radians (5.7°) of error, and coarse set under 5cm and 0.4 radians (22.9°). Related works (e.g. [4], [36]) often use this coarse threshold. For all experiments, we execute the controller until the tight threshold is realized, then switch to the next goal pose. In post-processing, we back-compute the time-to-goal under coarse thresholds (Table I).

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Mean $\pm \sigma$ [Min, Max]	Time to Goal (s) within Pose Tolerances	
	Tight: 2cm, 0.1rad	Loose: 5cm, 0.4rad
HW 3D Jack (Ours) 67 success (0 HWV), 0 fail	109.20 \pm 64.24 [17.40, 292.07]	84.86 \pm 60.54 [5.20, 257.74]
Sim 3D Jack (Ours) 26 success (0 HWV), 0 fail	49.31 \pm 30.35 [9.71, 124.70]	33.84 \pm 26.39 [7.97, 97.82]
Sim 3D Jack (Naive-R) 32 success (0 HWV), 18 fail	182.34 \pm 84.66 [63.52, 352.75]	138.17 \pm 78.89 [18.64, 297.00]
Sim 3D Jack (Naive-B) 3 success (0 HWV), 17 fail	140.86 \pm 121.46 [18.55, 306.47]	49.67 \pm 32.25 [17.69, 93.82]
Sim 3D Jack (MJPC) 34 success (25 HWV), 0 fail	107.91 \pm 112.38 [3.30, 567.69]	68.00 \pm 83.50 [1.51, 343.79]
HW Planar Push-T (Ours) 106 success (0 HWV), 0 fail	30.45 \pm 13.11 [7.50, 79.43]	17.43 \pm 7.59 [3.86, 42.00]

TABLE I

PERFORMANCE METRICS OF HARDWARE (HW) AND SIMULATION (SIM) EXPERIMENTS UNDER TIGHT AND LOOSE TOLERANCES. HW LIMIT VIOLATIONS (HWV) INDICATED. MJPC RESULTS USE HIGHEST-PERFORMING END EFFECTOR VELOCITY COST OF 0.09.

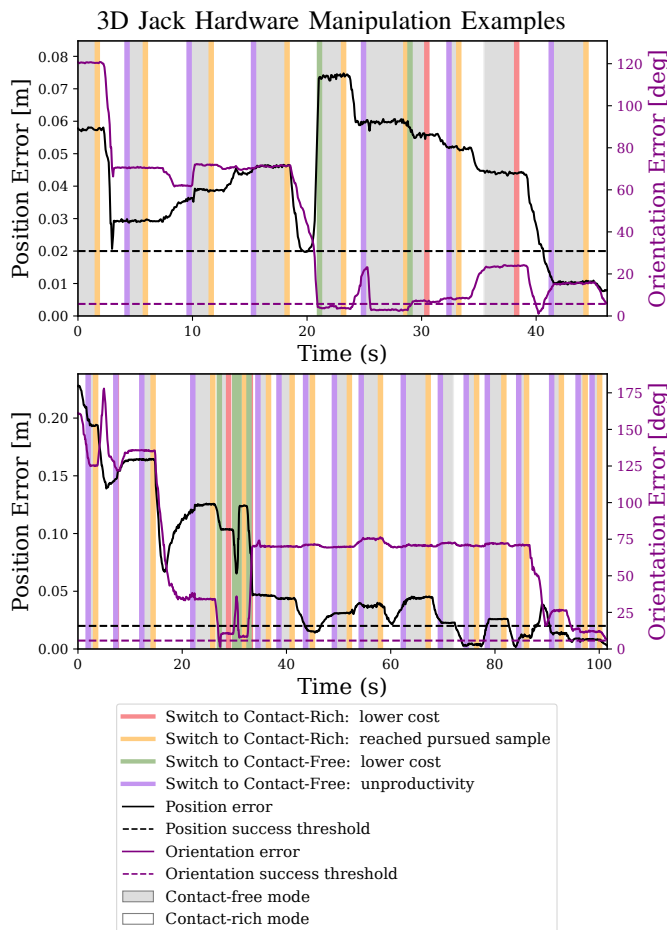


Fig. 9. Jack position and orientation errors over time, with contact-free (grey) and contact-rich (white) modes shaded and mode switching reasons labeled.

A. 3D Jack

Fig. 7 combines hardware results from four continuous experiments of 21, 16, 15, and 15 achieved random pose goals. All experiments terminated when the robot hit workspace safety limits. The time-to-goal cumulative distribution shows

that while most goals are reached relatively quickly, there is a long tail with some challenging targets requiring more time.

Fig. 9 depicts two single-goal trials achieved on hardware. These show the controller’s balancing position and rotation progress, occasionally sacrificing one to make progress on the other. Both trials demonstrate cost- and progress-based mode transitions. Most often, errors do not change during contact-free mode, since the object usually stays fixed while the robot relocates. One cause of inefficiency is simultaneous low position error and high rotation error. The only terminal failure condition is when the object is pushed to the boundary of the safe workspace, causing the robot to cross our workspace safety limits. In all other trials, our controller’s persistence eventually brings the object to the goal in every test.

1) *Sim-to-Real Gap and Comparisons*: Our controller, unsurprisingly, achieves pose goals faster in simulation than on hardware (see 49.31s simulation average compared to 109.20s hardware average in Table I). This gap, common in the literature, can be partly explained by state estimate errors, incorrect models, and FoundationPose’s added computational load. We reiterate our approach is not optimal; it is a real-time CI-MPC approach that makes all its decisions on-the-fly. While not optimal, our controller is demonstrably effective on difficult tasks and outperforms MJPC [5] and our ablations in simulation on our 3D jack example (Table I, Fig. 8). Further, our controller and its ablation variants (Naive-R and Naive-B) satisfy Franka hardware limits (joint velocity/torque, workspace limits), while MJPC nearly always violates at least one. Given this high hardware limit violation (HWV) rate, we did not feel safe deploying MJPC on the real robot. While impossible to compare against all possible MJPC weights, we investigate the role of end effector velocity cost weight, due to its role in the controller’s speed. Increasing this cost weight makes MJPC less performant, yet does not prevent HWV.

For the majority of trials, Naive-B is incapable of bringing the object to the goal. Its failure mode is when the sample behind the goal is unhelpful, so no contact is established during the contact-rich mode. Thus the sample does not update in response to object movement, and the robot will continually switch between contact-free and contact-rich modes without further moving the object. In contrast, Naive-R (like our approach) should always bring the object to the goal, though requiring more time to do so, indicating using the CI-MPC cost is a useful identifier of more effective contact-rich modes.

B. Planar T Pushing

The statistics in Table I for the hardware planar T experiments combine four continuous experiments of 56, 20, 20, and 10 successfully achieved random planar pose goals in a row. With an average of 30.45s to achieve high-precision pose goals, our approach achieves a state-of-the-art time-to-goal competitive with other works on this example. Notably, while some prior works with this example use data-driven approaches (e.g. pre-training with Diffusion Policy [1]) or require offline computation (e.g. offline trajectory optimization [37]), our approach demonstrates only the object model is required, and generalization to different goal poses is a natural byproduct absent from these prior works.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

VII. LIMITATIONS

While we contribute a solution to the global, 3D manipulation problem, our controller still took ~ 1.8 minutes on average over 67 trials to achieve precise SE(3) goals. We reiterate the generality of our approach and acknowledge these tasks are challenging. However, addressing inefficiencies is future work. Like all model-based methods, our controller requires dynamics models of the robot, objects, and environment, preventing use in truly novel scenarios. Our demonstrations used one spherical end effector, which we reasonably modeled as a single robot-object contact pair. If applied to a more dexterous manipulator such as a multi-fingered hand, contact pairs and state size would increase, slowing down control rates.

VIII. CONCLUSION

Our model-based controller performs generalizable, precise pose-driven tasks through multi-contact dynamics. By splitting the control problem into contact-free and contact-rich stages, we reap the benefits of global exploration when we sample new end effector locations plus local efficacy when existing CI-MPC methods take over after arriving at a desired location. In closed loop, the result is a persistent, globally-aware controller that can robustly reach precise pose goals without intervention.

REFERENCES

- [1] 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*, p. 02783649241273668, 2023.
- [2] X. Cheng, S. Patil, Z. Temel, O. Kroemer, and M. T. Mason, "Enhancing dexterity in robotic manipulation via hierarchical contact exploration," *arXiv preprint arXiv:2307.00383*, 2023.
- [3] T. Pang, H. 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*, 2023.
- [4] A. H. Li, P. Culbertson, V. Kurtz, and A. D. Ames, "Drop: Dexterous reorientation via online planning," *arXiv preprint arXiv:2409.14562*, 2024.
- [5] T. Howell, N. Gileadi, S. Tunyasuvunakool, K. Zakka, T. Erez, and Y. Tassa, "Predictive sampling: Real-time behaviour synthesis with mujoco," *arXiv preprint arXiv:2212.00541*, 2022.
- [6] W. Heemels, J. M. Schumacher, and S. Weiland, "Linear complementarity systems," *SIAM journal on applied mathematics*, vol. 60, no. 4, pp. 1234–1269, 2000.
- [7] A. Patel, S. L. Shield, S. Kazi, A. M. Johnson, and L. T. Biegler, "Contact-implicit trajectory optimization using orthogonal collocation," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2242–2249, 2019.
- [8] 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.
- [9] R. Antonova, J. Yang, K. M. Jatavallabhula, and J. Bohg, "Rethinking optimization with differentiable simulation from a global perspective," in *Conference on Robot Learning*. PMLR, 2023, pp. 276–286.
- [10] 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.
- [11] H. Zhu, A. Meduri, and L. Righetti, "Efficient object manipulation planning with monte carlo tree search," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 10 628–10 635.
- [12] L. Liu, K. Yin, M. Van de Panne, T. Shao, and W. Xu, "Sampling-based contact-rich motion control," in *ACM SIGGRAPH 2010 papers*, 2010, pp. 1–10.
- [13] B. Acetuno and A. Rodriguez, "A hierarchical framework for long horizon planning of object-contact trajectories," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 189–196.
- [14] M. A. Toussaint, K. R. Allen, K. A. Smith, and J. B. Tenenbaum, "Differentiable physics and stable modes for tool-use and manipulation planning," 2018.
- [15] J. Kamat, J. Ortiz-Haro, M. Toussaint, F. T. Pokorný, and A. Orthey, "Bitkomo: Combining sampling and optimization for fast convergence in optimal motion planning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4492–4497.
- [16] N. Chavan-Dafle and A. Rodriguez, "Sampling-based planning of in-hand manipulation with external pushes," in *Robotics Research: The 18th International Symposium ISRR*. Springer, 2019, pp. 523–539.
- [17] W. Yang and M. Posa, "Dynamic on-palm manipulation via controlled sliding," in *Robotics: Science and Systems (RSS)*, Jul. 2024. [Online]. Available: <https://roboticsconference.org/program/papers/12/>
- [18] A. Aydinoglu, A. Wei, W.-C. Huang, and M. Posa, "Consensus complementarity control for multi-contact mpc," *IEEE Transactions on Robotics*, 2024.
- [19] G. Kim, D. Kang, J.-H. Kim, S. Hong, and H.-W. Park, "Contact-implicit mpc: Controlling diverse quadruped motions without pre-planned contact modes or trajectories," *arXiv preprint arXiv:2312.08961*, 2023.
- [20] S. L. Cleac'h, T. Howell, M. Schwager, and Z. Manchester, "Fast contact-implicit model-predictive control," *arXiv preprint arXiv:2107.05616*, 2021.
- [21] V. Kurtz, A. Castro, A. Ö. Önel, and H. Lin, "Inverse dynamics trajectory optimization for contact-implicit model predictive control," *arXiv preprint arXiv:2309.01813*, 2023.
- [22] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 4906–4913.
- [23] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1433–1440.
- [24] H. Xue, C. Pan, Z. Yi, G. Qu, and G. Shi, "Full-order sampling-based mpc for torque-level locomotion control via diffusion-style annealing," 2024. [Online]. Available: <https://arxiv.org/abs/2409.15610>
- [25] D. E. Stewart, *Dynamics with Inequalities: impacts and hard constraints*. SIAM, 2011.
- [26] M. Halm and M. Posa, "Set-valued rigid-body dynamics for simultaneous, inelastic, frictional impacts," *The International Journal of Robotics Research*, p. 02783649241236860, 2024.
- [27] M. R. Dogar and S. S. Srinivasa, "A planning framework for non-prehensile manipulation under clutter and uncertainty," *Autonomous Robots*, vol. 33, pp. 217–236, 2012.
- [28] K. M. Lynch and M. T. Mason, "Stable pushing: Mechanics, controllability, and planning," *The international journal of robotics research*, vol. 15, no. 6, pp. 533–556, 1996.
- [29] M. Anitescu and F. A. Potra, "Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems," *Nonlinear Dynamics*, vol. 14, no. 3, pp. 231–247, 1997.
- [30] R. Tedrake *et al.*, "Drake: Model-based design and verification for robotics," 2019.
- [31] P. M. Wensing and D. E. Orin, "Generation of dynamic humanoid behaviors through task-space control with conic optimization," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3103–3109.
- [32] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 868–17 879.
- [33] A. S. Huang, E. Olson, and D. C. Moore, "Lcm: Lightweight communications and marshalling," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 4057–4062.
- [34] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "Osqp: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.
- [35] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2024. [Online]. Available: <https://www.gurobi.com>
- [36] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, and P. Agrawal, "Visual dexterity: In-hand reorientation of novel and complex object shapes," *Science Robotics*, vol. 8, no. 84, p. eadc9244, 2023.
- [37] S. Kang, G. Liu, and H. Yang, "Global contact-rich planning with sparsity-rich semidefinite relaxations," *arXiv preprint arXiv:2502.02829*, 2025.