Global Planning for Contact-Rich Manipulation via Local Smoothing of Quasi-dynamic Contact Models

Tao Pang*, H.J. Terry Suh*, Lujie Yang and Russ Tedrake

***Abstract*—The empirical success of Reinforcement Learning (RL) in contact-rich manipulation leaves much to be understood from a model-based perspective, where the key difficulties are often attributed to (i) the explosion of contact modes, (ii) stiff, non-smooth contact dynamics and the resulting exploding / discontinuous gradients, and (iii) the non-convexity of the planning problem. The stochastic nature of RL addresses (i) and (ii) by effectively sampling and averaging the contact modes. On the other hand, model-based methods have tackled the same challenges by smoothing contact dynamics analytically. Our first contribution is to establish the theoretical equivalence of the two smoothing schemes for simple systems, and provide qualitative and empirical equivalence on several complex examples. In order to further alleviate (ii), our second contribution is a convex, differentiable and quasi-dynamic formulation of contact dynamics, which is amenable to both smoothing schemes, and has proven to be highly effective for contact-rich planning. Our final contribution resolves (iii), where we show that classical sampling-based motion planning algorithms can be effective in global planning when contact modes are abstracted via smoothing. Applying our method on several challenging contact-rich manipulation tasks, we demonstrate that efficient model-based motion planning can achieve results comparable to RL, but with dramatically less computation.**

https://sites.google.com/view/global-planning-contact/home

***Index Terms*—Manipulation Planning, Dexterous Manipulation, Motion and Path Planning, Contact Modeling.**

I. Introduction

RECENT advances in RL have shown impressive results in manipulation through contact-rich dynamics that were difficult to achieve with previous model-based methods [1]–[3]. However, it is yet unclear what made these methods successful where model-based methods have struggled. Our high-level goal is to understand these reasons and interpret them from a model-based point of view. From such interpretations, we seek to apply the ingredients behind the empirical success of RL with the generalizability and efficiency of models.

From a model-based perspective, the most significant obstacle for planning through contact lies in the *hybrid* nature of contact dynamics (i.e. numerous modes of smooth dynamics separated by guard surfaces). The non-smooth nature of the resulting dynamics implies that Taylor approximation no longer holds locally; thus, the locally linear model constructed with the gradient quickly becomes invalid. This invalidity of the local model presents significant challenges for both iterative gradient-based optimization [4], [5], and sampling-based planning that rely on local distance metrics [6]–[8].

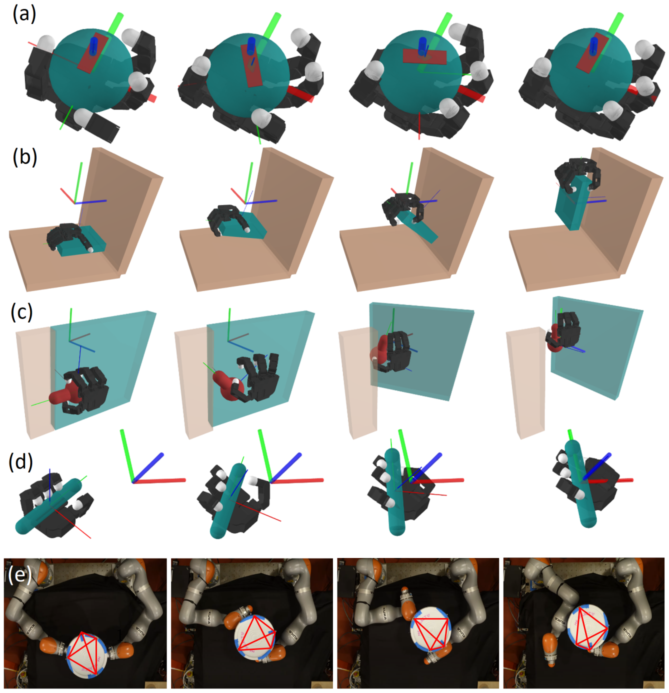


Fig. 1: Examples of contact-rich plans generated by our method. Each row corresponds to a time-snapshot of five different tasks: (**a**) 3D in-hand manipulation, (**b**) plate pickup using extrinsic dexterity, (**c**) door opening with a hinge and (**d**) pen placement, (**e**) bucket rotation with two iiwa arms on hardware. Each plan is generated online in the order of a minute of wall-clock time.

Faced with such challenges, many existing works have sought to explicitly consider contact modes by either enumerating or sampling them. With model-based knowledge of the dynamic modes, these planners typically alternate between continuous-state planning in the current contact mode and a discrete search for the next mode, resulting in trajectories punctuated by a handful of mode changes [7], [9], [10]. Another example of such approaches comes from transcribing the modes into binary variables, resulting in Mixed-Integer Programs [11]–[13]. However, both of these methods suffer from poor scalability: complex systems such as 3D in-hand manipulation can have trillions of contact modes [14]. The explosion of contact modes indicates that a viable planning strategy in the regime of rich contacts needs to somehow avoid considering all possible contact sequences [15].

Indeed, recent works have attributed the success of RL to its stochastic nature, where the contact modes are stochastically abstracted [16]–[18] by a process of sampling and averaging.

*These authors contributed equally to this work.

This work is funded by Amazon PO 2D-06310236, Lincoln Laboratory/Air Force PO 7000470769, NSF Award No. EFMA-1830901 and the Ocado Group. The authors are with the Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology, USA. {pangtao, hjsuh94, lujie, russt}@csail.mit.edu

This procedure, termed *randomized smoothing*, stochastically smooths the underlying landscape [19]. These works show that randomized smoothing can locally abstract neighboring contact modes into a single average linear model, implicitly creating a *stochastic force field* where contact is applied from a distance by means of randomization. This average model can give more informative gradients that are useful for planning.

However, the idea of smoothing contact modes for planning is hardly new in model-based literature. Many of the previous works in planning through contact have employed schemes to iteratively smooth contact dynamics in order to improve convergence of the solvers [15], [20]–[23], which leads to similar force-from-a-distance interpretations. Importantly, these methods differ from randomized smoothing as they consider the structure of the underlying equations to produce smooth approximations. We term this process *analytic smoothing*. The presence of the two smoothing schemes, randomized and analytic, begs an important question: what differences, both in terms of theoretical characterizations and empirical performance, can we expect from both smoothing schemes?

Our first contribution is to establish the theoretical equivalence of the two smoothing schemes for simple systems under our framework (Sec.II,IV-C). Using this framework, we further show how to efficiently compute the locally linear models (i.e. gradients) of the smoothed dynamics online (Sec.II-C), and show that both smoothing schemes result in comparable qualitative characteristics, as well as empirical performance on a number of complex examples (Sec.V,VIII)

Furthermore, at the heart of the model-based planning approach we take lies the question: what is the *right* model for gradient-based contact-rich manipulation planning? We believe this model needs to be (**i**) numerically robust and (**ii**) differentiable, so that local linearizations can be efficiently computed for planning. Most importantly, the model should be able to (**iii**) predict *long-term* behavior, so that the planner can look far ahead while taking few steps. Lastly, the model should be (**iv**) amenable to smoothing in order to provide more informative gradients across contact modes.

Our second contribution (Sec.III) is a contact dynamics model that satisfies all requirements. Specifically, we propose a *convex* implicit time-stepping contact model (Sec.III-B). Convexity is achieved by Anitescu's relaxation of frictional contact constraints [24], which in practice only introduces mild non-physical behavior [25], [26]. The convexity provides a clear numerical advantage over the traditional Linear Complementarity Problem (LCP) formulation [27], [28]. Moreover, the derivatives of the dynamics with respect to the current state and control action can be readily computed using sensitivity analysis for convex conic programs [29] (Sec.III-C).

For long-term predictability, we adopt the quasi-dynamic assumption widely used in robotic manipulation [10], [11], [30], [31]. A quasi-dynamic model "sees" further than its second-order counterpart by ignoring transient dynamics and focuses on transitions between steady-states. Furthermore, by throwing away kinetic energy at every time step, quasi-dynamic models are also simpler as they do not need variables for velocity, nor parameters to describe damping.

We validate and test our quasi-dynamic contact model by running the same input trajectories in a high-fidelity second-order simulator Drake [32], as well as on hardware (Sec.IX). Our results show that our model is able to approximate the second-order dynamics well if the system considered is highly damped and truly dominated by frictional forces.

Our contact model can also be analytically smoothed out using a log-barrier relaxation (Sec.IV-B). In this relaxation scheme, the hard contact constraints are softly enforced by a log-barrier function, a common technique used in the interior-point method for convex programs [33, §11] [23]. We further show that the gradients of the smoothed contact model can be easily computed with the implicit function theorem.

Finally, we believe that another key factor behind the empirical success of RL lies in its goal of performing global optimization with stochasticity. In contrast, deterministic model-based optimization for planning through nonlinear dynamics generally results in non-convex optimization problems, where the quality of different local minima can be a make-or-break factor. Solving such problems requires non-trivial initialization and tuning [34], both of which can be highly problem specific and notoriously hard to debug. This motivates us to search further for a more global approach.

In robotics, sampling-based motion planning (SBMP) algorithms such as the Rapidly-Exploring Random Tree (RRT) [35] are widely used for global search problems, including those with kinodynamic constraints [36]. However, the success of such planners has rarely extended to contact-rich settings. We find that while optimization-based methods for planning through contact have employed smoothing schemes, all existing SBMP methods for contact planning explicitly consider modes instead of smoothing them [7], [9], [10], [31], [37], as SBMP methods do not inherently require local characterizations of dynamics (i.e. gradients). Yet, previous works have shown that such local models are highly relevant for designing more efficient distance metrics during the nearest neighbor queries that respects dynamic reachability [6]–[8].

Our final contribution is to fill in this gap by combining smoothing-based contact mode abstraction and the global search capabilities of RRT. We enable RRT to search through contact dynamics constraints by utilizing a novel distance metric derived from the local smoothed models (Sec.VI). In addition, we propose an efficient extension step by computing actions that expand the tree using the local models (Sec.VII). With a variety of contact-rich tasks inspired by [38] that involve both intrinsic and extrinsic dexterity [39], we show that combining smoothing with RRT achieves tractable global motion planning for highly contact-rich manipulation (Sec.VIII). To the best of our knowledge, our work appears to be the first to successfully combine SBMP with contact mode smoothing.

Summary of Contributions. (**i**) We establish the theoretical equivalence of methods for randomized and analytic smoothing on simple systems, and qualitative / empirical equivalence on complex systems. (**ii**) We present a convex, differentiable formulation of quasi-dynamic contact dynamics and its analytic smoothing, which we show to be highly effective for contact-rich manipulation planning. (**iii**) We combine contact mode smoothing with sampling-based motion planning, filling in a gap in the spectrum of existing methods and achieving efficient global planning through highly-rich contact dynamics.

II. Local Theory of Smoothing

Before discussing complex systems with contact, we formalize mathematically what it means to smooth a function, as well as different algorithms to compute their local approximations. We aim to provide a coherent view of the different smoothing schemes and their equivalence relations.

Consider a locally-Lipschitz (but potentially non-smooth) function $f : \mathbb{R}^n \to \mathbb{R}^m$. We consider a class of iterative algorithms that rely on finding a locally-affine approximation to f at every iteration, around the current operating point $\bar{x}$. We denote this affine model as $g(\cdot;\bar{x})$,

$$g(x;\bar{x}) \coloneqq \mathbf{J}(\bar{x})\underbrace{(x-\bar{x})}_{=:\delta x}+\mu(\bar{x}). \tag{1}$$

which consists of the sensitivity term $\mathbf{J}(\bar{x}) \in \mathbb{R}^{m\times n}$, and a bias term $\mu(\bar{x}) \in \mathbb{R}^m$, which we refer to as model parameters $(\mathbf{J}(\bar{x}), \mu(\bar{x}))$.

If $f \in C^1$ (i.e. smooth), then a natural candidate to use for the model parameters is the first-order Taylor expansion of f around $\bar{x}$, also known as the *linearization* of the function. The linearization requires obtaining the gradient of f.

Definition 1. We define the linearization of f around the nominal point $\bar{x}$ as $\bar{f}$,

$$\bar{f}(x;\bar{x}) \coloneqq \left.\frac{\partial f}{\partial x}\right|_{x=\bar{x}}(x-\bar{x}) + f(\bar{x}). \tag{2}$$

Note that $\bar{f}$ is a locally-affine model with parameters $(\partial f/\partial x|_{x=\bar{x}}, f(\bar{x}))$.

However, when f is non-smooth with fast-changing discontinuous gradients, the Taylor expansion (2) quickly deviates from f. Therefore, approaches that utilize (2) (e.g. approaches that use the gradient) often suffer greatly. In this section, we investigate a set of smoothing methods that can give more informative locally-affine approximations to f.

A. Smooth Surrogate of a Function

We first formalize the process of smoothing a function f by introducing the *smooth surrogate*, which is a modification of f that closely resembles the original function, yet is smooth.

1) The Smooth Surrogate: We define a smooth surrogate using the process of convolution.

Definition 2. Given a probability distribution ρ characterized by a smooth density function, we define f_ρ as the smooth surrogate of f,

$$f_\rho(x) \coloneqq \mathbb{E}_{w\sim\rho}[f(x+w)], \tag{3}$$

The smooth surrogate provides us with a natural formalism of common smooth approximations that are made to non-smooth functions. We give some examples of smooth surrogates that are commonly used in machine learning and robotics, as well as their corresponding densities, in Table I.

Note that the expectation (3) is applied pointwise. Given a single point x, it is desirable to average points from a neighborhood that is centered at x. Thus, we make the following assumption on the distribution ρ, which we further assume to be smooth and elliptical to enforce regularity.

f	ρ	f_ρ
ReLU	Logistic	SoftPlus
Heaviside	Cauchy	ArcTan
Heaviside	Logistic	Sigmoid

TABLE I: Examples of smooth surrogates of non-smooth functions.

Assumption 1. We assume ρ is a zero-mean elliptical distribution with an integrable and smooth density: $\rho \in L^1, \rho \in C^1$.

The smoothness of f_ρ, despite the non-smoothness of f, is formalized as the following lemma which follows from properties of convolution.

Lemma 1. If $\rho \in C^1$, then $f_\rho \in C^1$.

Proof. We show that the gradients are well-defined using Leibniz integral rule,

$$\begin{aligned} \frac{d}{dx}f_\rho(x) &= \tfrac{d}{dx}\textstyle\int f(x+w)\rho(w)dw \\ &= \tfrac{d}{dx}\textstyle\int f(y)\rho(y-x)dy \\ &= \textstyle\int f(y)\left(\tfrac{d}{dx}\rho(y-x)\right)dy, \end{aligned} \tag{4}$$

where we make the reparametrization $y = x+w$, and $\frac{d}{dx}\rho(y-x)$ is well-defined since $\rho \in C^1$. □

2) Linearization of the Smooth Surrogate: Since $f_\rho \in C^1$, first-order Taylor approximation now holds for f_ρ and we can successfully use the linearization of f_ρ as an affine model that best represents f_ρ at an operating point of $\bar{x}$.

Definition 3. We apply the definition of linearization on f_ρ to define the linearization of the smooth surrogate,

$$\begin{aligned} \mathbf{J}_\rho(\bar{x}) &\coloneqq \tfrac{\partial}{\partial x}f_\rho(\bar{x}) = \tfrac{\partial}{\partial x}\mathbb{E}_{w\sim\rho}[f(x+w)]|_{x=\bar{x}} \\ \mu_\rho(\bar{x}) &\coloneqq f_\rho(\bar{x}) = \mathbb{E}_{w\sim\rho}[f(\bar{x}+w)]. \end{aligned} \tag{5}$$

We note that there are two alternative ways to characterize $\mathbf{J}_\rho$. First, the following lemma establishes the validity of the *reparametrization gradient* of f_ρ [40], which requires access to gradients of f.

Lemma 2. $\mathbf{J}_\rho$ can be obtained by taking the expectation of gradients,

$$\mathbf{J}_\rho(\bar{x}) = \mathbb{E}_{w\sim\rho}\left[\left.\tfrac{\partial}{\partial x}f(x)\right|_{x=\bar{x}+w}\right]. \tag{6}$$

Proof. The exchange of derivative and expectation follows from Leibniz integral rule. □

The next lemma establishes the validity of the *score function* gradient of f_ρ, which does not requires access to gradients of f. This gradient, commonly used in RL, is also referred to as the likelihood ratio, or the REINFORCE gradient [41].

Lemma 3. (**Stein's Lemma**) For any choice of b that does not depend on w, $\mathbf{J}_\rho$ can be obtained as follows,

$$\mathbf{J}_\rho(\bar{x}) = \mathbb{E}_{w\sim\rho}\left[[f(\bar{x}+w)-b]\underbrace{\left[-\frac{\nabla\rho(w)}{\rho(w)}\right]^\intercal}_{=:S(w)}\right]. \tag{7}$$

Proof. We refer to [42], [43] for detailed proof for the case that $b = 0$. If b is non-zero, we use the fact that the expectation of the score function $S(w)$ is zero, $\mathbb{E}_w S(w) = 0$ to show that the choice of b has no impact on the equality. □

B. Rethinking Linearization as a Minimizer

We also provide an alternative motivation of smoothing. Instead of interpreting smoothing as modifying f directly, we can provide an alternative interpretation of linearization: what is the *best* affine approximation of f around a nominal point $\bar{x}$, and according to which metric? We propose a natural objective of the residual between the original function f and the linear model g, weighted by some distribution ρ. Since f_ρ is an affine model that has to minimize the residual from a distribution rather than a single point, we expect the affine model to be more informative compared to the exact linearization, especially when f is non-smooth. We formalize this through the following theorem.

Theorem 1. Let ρ be a zero-mean Gaussian with covariance $\mathbf{\Sigma}$, $\rho(w) = \mathcal{N}(w; 0, \mathbf{\Sigma})$. Consider the problem of choosing the best affine model with parameters $(\mathbf{J}, \mu)$ such that the residual distributed according to ρ is minimized,

$$\min_{\mathbf{J},\mu} \frac{1}{2}\mathbb{E}_{w\sim\rho}\left[\|f(\bar{x}+w) - \mathbf{J}w - \mu\|_2^2\right]. \tag{8}$$

Then, the solution to the problem is given by the exact linearization of the smooth surrogate,

$$\mathbf{J}^* = \mathbf{J}_\rho(\bar{x}), \quad \mu^* = \mu_\rho(\bar{x}). \tag{9}$$

Proof. (8) is a linear regression problem and convex in the arguments, and thus, the first-order stationarity condition implies optimality. Letting $F(\mathbf{J}, \mu)$ denote the objective function, the gradients are

$$\partial F/\partial \mu = \mathbb{E}_w[f(\bar{x}+w)] - \mu^*, \tag{10a}$$

$$\partial F/\partial \mathbf{J} = \mathbb{E}_w[ww^\intercal]\mathbf{J}^* - \mathbb{E}_w[f(\bar{x}+w)w^\intercal]. \tag{10b}$$

Setting them to zero, (10a) directly leads to the solution for μ^*. In addition, $\mathbf{J}^*$ can be obtained from (10b):

$$\begin{aligned} \mathbf{J}^* &= \mathbb{E}_{w\sim\rho}[ww^\intercal]^{-1}\mathbb{E}_{w\sim\rho}[f(\bar{x}+w)w^\intercal] \\ &= \tfrac{\partial}{\partial x}\mathbb{E}_{w\sim\rho}[f(x+w)]|_{x=\bar{x}}, \end{aligned} \tag{11}$$

which comes from using Stein's lemma, where the score function of the Gaussian is $S(w) = \mathbf{\Sigma}^{-1}w$. □

Theorem 1 offers another insight of why smoothing is necessary if we generalize the objective around a distribution. A slightly technical modification of this theorem, which replaces $\mathbf{J}w$ in (8) with $\mathbf{J}\mathbf{F}^{-1}(w)S(w)$ where $\mathbf{F}(w)$ is the Fisher information matrix $\mathbf{F}(w) := \mathbb{E}_{w\sim\rho}[S(w)S(w)^\intercal]$, also applies for general elliptical distributions with identical proof. However, we document the Gaussian case for intuition.

Example 1. To illustrate the different smoothing schemes and the implication of Theorem 1, we plot a simple example for a polynomial f in Fig. 2. The example shows that when least-squares fit is performed on the samples of $f(\bar{x}+w_i)$, the linear model that minimizes (8) converges to the exact linearization of the smooth surrogate f_ρ.

C. Computation of the Locally Linear Model

The equivalence between different characterizations of $(\mathbf{J}_\rho, \mu_\rho)$ implies different methods to compute them with access to f. We investigate three major methods to compute these quantities.

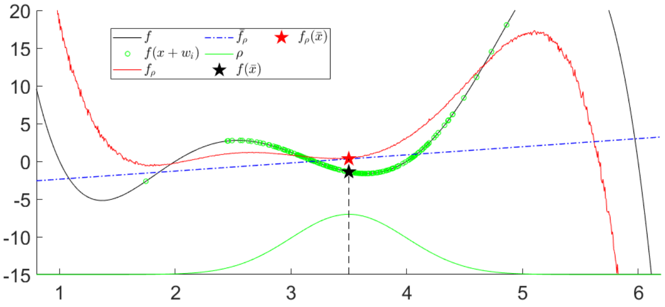


Fig. 2: Polynomial f (black) and its smooth surrogate f_ρ (red), for the case where ρ is a Gaussian (green). The plotted f_ρ is obtained using Monte-Carlo estimation. When $f(\bar{x}+w_i)$ (green) are sampled around $\bar{x}$ and the least squares fit is performed (blue line, $\bar{f}_\rho$), it converges to the exact linearization of the smooth surrogate f_ρ.

1) Analytical Smoothing : If f and ρ are sufficiently structured, one can analytically compute the convolution f_ρ and its derivative $\mathbf{J}_\rho$, similar to the analytic derivatives of f_ρ on simple functions from Table I.

2) Randomized Smoothing, First-Order : If ρ is a distribution that is easy to sample from, one can compute a Monte-Carlo estimate of μ_ρ by averaging the samples.

Lemma 4. The sample mean of $f(x+w_i)$, where w_i are drawn i.i.d. from ρ, is an unbiased estimator of μ_ρ,

$$\mathbb{E}_{w_i \overset{\text{i.i.d.}}{\sim} \rho}\left[\frac{1}{N}\sum_{i=1}^N f(\bar{x}+w_i)\right] = \mu_\rho(\bar{x}). \tag{12}$$

Proof. Law of large numbers. □

Furthermore, Lemma 2 implies that $\mathbf{J}_\rho$ can be estimated by sampling the gradients. We note that crucially, use of this method requires f to be locally-Lipschitz (i.e. continuous). The failure mode of first-order randomized smoothing is documented in [16], [17], [44] and also briefly illustrated in Example 2.

Lemma 5. Under locally Lipschitz f, the sample mean of $\frac{\partial f}{\partial x}(x+w_i)|_{x=\bar{x}}$, where w_i are drawn i.i.d. from ρ, is an unbiased estimator of $\mathbf{J}_\rho$,

$$\mathbb{E}_{w_i \overset{\text{i.i.d.}}{\sim} \rho}\left[\frac{1}{N}\sum_{i=1}^N \frac{\partial f}{\partial x}(\bar{x}+w_i)\right] = \mathbf{J}_\rho(\bar{x}). \tag{13}$$

Proof. We use Law of large numbers then Lemma 2. □

3) Randomized Smoothing, Zeroth-Order: Finally, Lemma 3 implies that we can estimate $\mathbf{J}_\rho$ in zeroth-order, which is analogous to how gradients are computed in RL [17], [41].

Lemma 6. For any choice of b that does not depend on w, we have

$$\mathbb{E}_{w_i \overset{\text{i.i.d.}}{\sim} \rho}\left[\frac{1}{N}\sum_{i=1}^N [f(\bar{x}+w_i) - b]\, S(w_i)\right] = \mathbf{J}_\rho(\bar{x}). \tag{14}$$

Proof. We use Law of large numbers then Lemma 3. □

A typical choice of b is the zero-noise evaluation of f, $b = f(\bar{x})$. Note that if ρ is Gaussian, we can also take a finite-sample approximation to the least-squares problem:

$$\min_{\mathbf{J}_\rho, \mu_\rho} \frac{1}{N} \sum_{i=1}^{N} \|f(\bar{x} + w_i) - \mathbf{J}_\rho w_i - \mu_\rho\|_2^2 . \tag{15}$$

Unlike the first-order estimator, the zeroth-order estimator is still unbiased for discontinuous functions [17]. However, the variance of this estimator is often higher than that of the first-order estimator used in Lemma 2. This effect is also illustrated in Example 2.

Example 2. (**ReLU, Heaviside, Delta**). To better illustrate different computational methods and their properties, we revisit an example from Table I, where f is the ReLU function and ρ is a logistic distribution, for which the smooth surrogagte f_ρ is the Softplus.

We illustrate the three different methods, introduced in Sec.II-C, to compute the gradient of f_ρ. In the analytic scheme, we simply differentiate softplus to obtain a sigmoid function. The first-order randomized version averages the gradients $\partial f/\partial x$ (a Heaviside) with the samples drawn from ρ. Finally, the zeroth-order version uses Stein's lemma for the logistic distribution, which has the score function $S(w) = \tanh(w)$. The results are plotted in Fig.3. Note that both quantities converge to the analytic gradient on average, with the zeroth-order gradient having more variance than the first-order one.

Lastly, given $g := \partial f/\partial x$ as the Heaviside, the smooth surrogate g_ρ becomes the Sigmoid. We illustrate the three methods of computing the gradient of g_ρ, which is equivalent to $d^2 f_\rho/dx^2$. Using the analytic scheme, we recover the density function for the logistic distribution, and the zeroth-order estimator is also successful in recovering the gradient. However, as the gradients of the Heaviside (i.e. dirac-delta) is almost-surely zero, the first-order estimator becomes biased and fails to estimate the gradient of g_ρ.

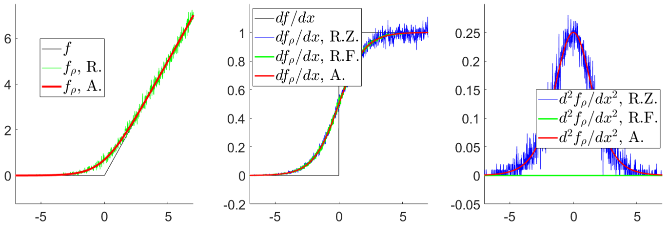


Fig. 3: A. analytic, R.F. randomized first, R.Z. randomized zero. Left: ReLU f (black), its smooth surrogate f_ρ, the softplus (red), and the Monte-Carlo approximation of f_ρ (green). Middle: Gradients of f_ρ computed using different methods. Right: Gradient of g_ρ, which we denote as $d^2 f_\rho/dx^2$, computed by different computation methods. Color scheme is identical.

Finally, we note that in the context of randomized smoothing, the first-order and the zeroth-order gradient estimators can be combined and interpolated, leading to a α-ordered gradient estimator where $\alpha \in [0, 1]$ [17]. Such interpolation smoothly trades off the bias and variance characteristics of the first and zeroth-order gradients.

D. Smoothing of Dynamical Systems

Our theorems in the setting of functions readily extends to dynamical systems. We overload the notation on the variables for convention, and say the system has state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^m$, and map $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$,

$$x_+ = f(x, u). \tag{16}$$

where we use $(\mathbf{A}(\bar{x}, \bar{u}), \mathbf{B}(\bar{x}, \bar{u}), c(\bar{x}, \bar{u}))$ to parameterize the linear system $\bar{f}$ that best describes f around some nominal point $(\bar{x}, \bar{u})$,

$$\bar{f}(x, u) = \mathbf{A}(\bar{x}, \bar{u}) \underbrace{(x - \bar{x})}_{=:\delta x} + \mathbf{B}(\bar{x}, \bar{u}) \underbrace{(u - \bar{u})}_{=:\delta u} + c(\bar{x}, \bar{u}). \tag{17}$$

For dynamical systems, the Taylor expansion requires the system Jacobian,

$$\begin{aligned} \mathbf{A}(\bar{x}, \bar{u}) &= \frac{\partial}{\partial x} f(x, u)|_{x=\bar{x}, u=\bar{u}} \\ \mathbf{B}(\bar{x}, \bar{u}) &= \frac{\partial}{\partial u} f(x, u)|_{x=\bar{x}, u=\bar{u}} \\ c(\bar{x}, \bar{u}) &= f(\bar{x}, \bar{u}). \end{aligned} \tag{18}$$

Similarly in this setting, the smooth surrogate can be defined as the surrogate dynamics

$$f_\rho(x, u) = \mathbb{E}_{w \sim \rho}[f(x + w^x, u + w^u)], \tag{19}$$

where w^x is the component of w that corresponds to x, and w^u is defined similarly. The model parameters of the locally linear model $\bar{f}_\rho$ are given as follows:

$$\begin{aligned} \mathbf{A}_\rho(\bar{x}, \bar{u}) &= \frac{\partial}{\partial x} \mathbb{E}_{w \sim \rho}[f(x + w^x, u + w^u)]|_{x=\bar{x}, u=\bar{u}} \\ \mathbf{B}_\rho(\bar{x}, \bar{u}) &= \frac{\partial}{\partial u} \mathbb{E}_{w \sim \rho}[f(x + w^x, u + w^u)]|_{x=\bar{x}, u=\bar{u}} \\ c_\rho(\bar{x}, \bar{u}) &= \mathbb{E}_{w \sim \rho}[f(\bar{x} + w^x, \bar{u} + w^u)]. \end{aligned} \tag{20}$$

The previous theorems (Lemma 2, Lemma 3, Theorem 1) also extend to this setting with slight changes in notation. In the remaining sections, we shorthand the notation and refer to the model parameters as a matrix instead of a function for compactness (e.g. $\mathbf{A}_\rho$ instead of $\mathbf{A}_\rho(\bar{x}, \bar{u})$).

1) Analytic Smoothing: If f is sufficiently structured, we can analytically obtain f_ρ as well as the parameters for its linearization $(\mathbf{A}_\rho, \mathbf{B}_\rho, c_\rho)$ without a sampling procedure.

2) First-Order Randomized Smoothing: We use the following estimators for first-order randomized smoothing of dynamical systems,

$$\mathbf{A}_\rho \approx \tfrac{1}{N} \textstyle\sum_{i=1}^{N} \left[\tfrac{\partial}{\partial x} f(\bar{x} + w_i^x, \bar{u} + w_i^u\right] \quad w_i \sim \rho \tag{21a}$$

$$\mathbf{B}_\rho \approx \tfrac{1}{N} \textstyle\sum_{i=1}^{N} \left[\tfrac{\partial}{\partial u} f(\bar{x} + w_i^x, \bar{u} + w_i^u))\right] \quad w_i \sim \rho \tag{21b}$$

$$c_\rho \approx \tfrac{1}{N} \textstyle\sum_{i=1}^{N} \left[f(\bar{x} + w_i^x, \bar{u} + w_i^u))\right] \quad w_i \sim \rho \tag{21c}$$

where $\partial f/\partial x$, and $\partial f/\partial u$ are Jacobians of the dynamics.

3) Zeroth-Order Randomized Smoothing: Similar to (15), we can estimate the gradients using least-squares when ρ is Gaussian:

$$\begin{aligned} \mathbf{A}_\rho, \mathbf{B}_\rho = \operatorname*{argmin}_{\mathbf{A}, \mathbf{B}} & \textstyle\sum_{i=1}^{N} \\ & \|f(\bar{x} + w_i^x, \bar{u} + w_i^u) - \mathbf{A} w_i^x - \mathbf{B} w_i^u - c_\rho\|_2^2 \end{aligned} \tag{22}$$

where c_ρ is computed with (21c).

III. Convex Quasi-dynamic Differentiable Contact Dynamics

In this section, we present the Convex, Quasi-dynamic, Differentiable Contact (CQDC) model which is used for the contact-rich planning tasks in the rest of this paper. We start by elaborating on the advantages of planning with quasi-dynamic models. We then detail the convex[1] forward dynamics formulation and how to take its derivatives, and conclude with remarks on implementation.

A. Why Plan with Quasi-dynamic Models?

Quasi-dynamic models, which previous works have used extensively for robotic manipulation [10], [11], [30], [31], [45], simplify Newtonian dynamics by removing terms related to velocity and acceleration, and focusing on contact forces at the core of robot-object interactions. Although the model cannot describe highly-dynamics behaviors such as spinning a pen between fingers [20], it holds up for a wide variety of manipulation tasks, including many that involve dexterous hands [1].

Halving the number of states by removing velocity is one of the most recognized advantages of quasi-dynamic systems [30, §10.1]. However, by focusing on transitions between static equilibria and ignoring transients induced by damping and acceleration, a quasi-dynamic model can predict further into the future while taking fewer steps than its second-order counterpart, thereby reducing the number of steps it requires to reach the goal. We believe this temporal aspect of quasi-dynamic models' advantage is important but often overlooked in the literature. In practice, many trajectory optimization formulations for manipulation enforce the second-order Newtonian dynamics [46], [47]. Although modeling the transients allows the discovery of more dynamic behaviors, we believe, especially in the manipulation setting, that the added computational complexity due to taking many steps for long planning horizons oftentimes outweighs the benefits.

B. Forward Dynamics

Despite the numerical advantage of convex contact dynamics formulations [24], [26], [48], some formulations can produce highly inaccurate contact forces with poorly-chosen hyperparameters [49]. In this work, we adopt Anitescu's convex relaxation of the Coulomb friction constraints [24]. Anitescu's convex relaxation is equivalent to common LCP formulations [28] in non-sliding contacts or in separation, and introduces a non-physical yet mild "boundary layer" effect between relatively-sliding objects [25], [26]. Not only does Anitescu's convex friction model have the simulation step size h as the only hyperparameter, the "boundary layer" also disappears as $h \to 0$. The step size therefore can serve as a knob that allows the planner to trade the dynamics model's predictive power for physical accuracy.

In the robotic manipulation setting, a system can be divided into n_a *actuated* Degree of Freedoms (DOFs), which correspond to the robots, and n_u *unactuated* DOFs, which correspond to the objects. The configurations of objects and robots are denoted respectively by $q^\mathrm{u} \in \mathbb{R}^{n_\mathrm{u}}$ and $q^\mathrm{a} \in \mathbb{R}^{n_\mathrm{a}}$. For quasi-dynamic systems, the system state is defined by $x := q := (q^\mathrm{u}, q^\mathrm{a})$. Similar to [25], we model the robots as impedances [50], which in the quasi-dynamic setting reduces to springs with a diagonal stiffness matrix $\mathbf{K}_\mathrm{a} \in \mathbb{R}^{n_\mathrm{a} \times n_\mathrm{a}}$. Accordingly, the input $u \in \mathbb{R}^{n_\mathrm{a}}$ is defined by the commanded positions of the robots' joints, which can also be interpreted as the equilibrium positions of the springs.

The discretized quasi-dynamic equations of motion are

$$h\mathbf{K}_\mathrm{a}\left(q^\mathrm{a} + \delta q^\mathrm{a} - u\right) = h\tau^\mathrm{a} + \sum_{i=1}^{n_\mathrm{c}} \left(\mathbf{J}_{\mathrm{a}_i}(q)\right)^\intercal \lambda_i, \tag{23a}$$

$$\left(\frac{\epsilon}{h}\mathbf{M}_\mathrm{u}(q)\right)\delta q^\mathrm{u} = h\tau^\mathrm{u} + \sum_{i=1}^{n_\mathrm{c}} \left(\mathbf{J}_{\mathrm{u}_i}(q)\right)^\intercal \lambda_i, \tag{23b}$$

where $h \in \mathbb{R}_{++}$ is the step size in seconds; $\mathbf{M}_\mathrm{u}(q) \in \mathbb{R}^{n_\mathrm{u} \times n_\mathrm{u}}$ is the configuration-dependent mass matrix of the objects; $\epsilon \in \mathbb{R}_+$ is a small regularization constant; $\tau^\mathrm{a} \in \mathbb{R}^{n_\mathrm{a}}$ and $\tau^\mathrm{u} \in \mathbb{R}^{n_\mathrm{u}}$ are non-contact external torques (e.g. due to gravity) for robots and objects; the change in system configuration from the current to the next step is $\delta q := (\delta q^\mathrm{u}, \delta q^\mathrm{a})$. For 3D systems, it is more common to work with angular velocity instead of rotational displacement. As our discussion trivially extends to angular velocity by using a linear map (e.g. [26, Section II]), we use displacement in this paper for brevity.

There are n_c contact pairs at the current time step. For the i-th contact, $\lambda_i \in \mathbb{R}^3$ is the contact impulse; $\mathbf{J}_{\mathrm{a}_i}(q) \in \mathbb{R}^{3 \times n_\mathrm{a}}$ and $\mathbf{J}_{\mathrm{u}_i}(q) \in \mathbb{R}^{3 \times n_\mathrm{u}}$ are the contact Jacobians [51] of the robots and the objects, respectively. The contact Jacobians are functions of q, as they depend on the contact points and normals at q computed by a collision detector. For brevity, we will drop the explicit dependency on q from here onward.

Equation (23a) states that the robot joint positions' deviation from their commanded values, as a result of external impulses, is proportional to the stiffness $\mathbf{K}_\mathrm{a}$. Equation (23b) can have different physical interpretations depending on the value of the regularization constant ϵ. If $\epsilon = 0$, (23b) is the force (impulse) balance of the objects. This corresponds to the exact quasi-static formulation in [25]. If $\epsilon = 1$, (23b) corresponds to Mason's classical definition of quasi-dynamic systems [30]. Note that for any ϵ, the momentum gained at every step due to external impulses is discarded at the next time step, which is characteristic of the highly damped behavior typical in quasi-static systems. In our physical experiments in later sections where objects are light, we find that the dynamics is closer to being exactly quasi-static ($\epsilon = 0$) than being classically quasi-dynamic ($\epsilon = 1$). Therefore, we pick ϵ to be as small as possible without causing numerical issues.

In addition, the Coulomb friction model requires the contact impulses λ_i to stay inside the friction cone, and the relative sliding velocities to satisfy the maximum dissipation principle [28]. To enforce these constraints, we first introduce some additional notation. The contact Jacobian for contact i can be written as

$$\mathbf{J}_i := \left[\mathbf{J}_{\mathrm{u}_i}, \mathbf{J}_{\mathrm{a}_i}\right] := \begin{bmatrix}\mathbf{J}_{\mathrm{n}_i} \\ \mathbf{J}_{\mathrm{t}_i}\end{bmatrix} \in \mathbb{R}^{3 \times (n_\mathrm{u} + n_\mathrm{a})}, \tag{24}$$

[1] By calling the forward dynamics $x_+ = f(x, u)$ convex, we mean that a convex optimization program is solved when computing x_+ from a given (x, u) pair. It does not mean that $f(x, u)$ is a convex function in (x, u).

where $\mathbf{J}_{\mathrm{n}_i} \in \mathbb{R}^{1\times(n_\mathrm{u}+n_\mathrm{a})}$ maps the generalized velocity of the system to the normal contact velocity, and $\mathbf{J}_{\mathrm{t}_i} \in \mathbb{R}^{2\times(n_\mathrm{u}+n_\mathrm{a})}$ to the tangent velocities. Next, the friction cone at contact i, $\mathcal{K}_i$, and its dual cone $\mathcal{K}_i^\star$, are denoted by

$$\mathcal{K}_i := \left\{\lambda_i = (\lambda_{\mathrm{n}_i}, \lambda_{\mathrm{t}_i}) \in \mathbb{R}^3 | \mu_i \lambda_{\mathrm{n}_i} \geq \sqrt{\lambda_{\mathrm{t}_i}^\intercal \lambda_{\mathrm{t}_i}}\right\}, \tag{25a}$$

$$\mathcal{K}_i^\star := \left\{v_i = (v_{\mathrm{n}_i}, v_{\mathrm{t}_i}) \in \mathbb{R}^3 | v_{\mathrm{n}_i} \geq \mu_i \sqrt{v_{\mathrm{t}_i}^\intercal v_{\mathrm{t}_i}}\right\}, \tag{25b}$$

where μ_i is the friction coefficient; the dual variable v_i can also be interpreted as the relative contact velocity for contact i; the subscripts $(\cdot)_\mathrm{n}$ and $(\cdot)_\mathrm{t}$ indicate respectively the normal and tangential components.

With this notation, Anitescu's frictional contact constraints can be written as:

$$v_i := \mathbf{J}_i \delta q + \begin{bmatrix}\phi_i \\ 0_2\end{bmatrix} \in \mathcal{K}_i^\star, \tag{26a}$$

$$\lambda_i \in \mathcal{K}_i, \tag{26b}$$

$$v_i^\intercal \lambda_i = 0, \tag{26c}$$

where $\phi_i \in \mathbb{R}$ is the signed distance for contact i at the current time step. These constraints enforce the Coulomb friction model exactly when a contact is sticking (not sliding) and in separation. In sliding, these constraints enforce maximum dissipation, but adds a small gap between the two relatively-sliding objects. Anitescu showed that this gap converges to 0 as $h \to 0$ [24].

Remarkably, this implies that the quasi-dynamic equations of motion (23), together with the friction constraints (26), are the KKT optimality conditions [33, §5.9] of the following convex Second-Order Cone Program (SOCP) [24]:

$$\min_{\delta q} . \ \frac{1}{2} \delta q^\intercal \mathbf{Q} \delta q + b^\intercal \delta q, \text{ subject to} \tag{27a}$$

$$\mathbf{J}_i \delta q + \begin{bmatrix}\phi_i \\ 0_2\end{bmatrix} \in \mathcal{K}_i^\star, \ \forall i \in \{1 \dots n_\mathrm{c}\}, \text{ where} \tag{27b}$$

$$\mathbf{Q} := \begin{bmatrix}\epsilon \mathbf{M}_\mathrm{u}/h & 0 \\ 0 & h\mathbf{K}_\mathrm{a}\end{bmatrix}, \ b := -h \begin{bmatrix}\tau^\mathrm{u} \\ \mathbf{K}_\mathrm{a}(u - q^\mathrm{a}) + \tau^\mathrm{a}\end{bmatrix}, \tag{27c}$$

whose primal and dual solutions, $\delta q^\star$ and $\lambda^\star := (\lambda_1^\star, \dots, \lambda_{n_\mathrm{c}}^\star)$, can be obtained by conic solvers such as [52], [53]. Note that without regularization ($\epsilon = 0$), the objective (27a) would be positive semi-definite, and thus (27) could have multiple solutions.

The SOCP (27) reduces to a Quadratic Program (QP) when there is no friction or if the friction cone can be described by linear constraints. The second-order friction cones (25) can be equivalently represented as linear constraints in the planar case, or be approximated with polyhedral cones [28] in the 3D case. However, this approximation introduces non-physical anisotropic behaviors [23], [54], which is why (25) is preferred.

C. Derivatives of Forward Dynamics

We illustrate how to compute the derivatives of the system configuration at the next step, q_+, with respect to the current q and u. Let us express the forward contact dynamics in the standard dynamical system form (16):

$$q_+ = f(q, u) = q + \delta q^\star(q, u), \tag{28}$$

where $\delta q^\star$ is the solution to (27). Taking the derivatives of (28) yields

$$\mathbf{A} = \frac{\partial f}{\partial q} = \mathbf{I} + \frac{\partial \delta q^\star}{\partial q}, \ \mathbf{B} = \frac{\partial f}{\partial u} = \frac{\partial \delta q^\star}{\partial u}, \tag{29}$$

where $\partial \delta q^\star / \partial q$ and $\partial \delta q^\star / \partial u$ can be expanded using the chain rule into:

$$\frac{\partial \delta q^\star}{\partial q} = \frac{\partial \delta q^\star}{\partial b} \frac{\partial b}{\partial q} + \frac{\partial \delta q^\star}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial q} + \sum_{i=1}^{n_\mathrm{c}} \frac{\partial \delta q^\star}{\partial \mathbf{J}_i} \frac{\partial \mathbf{J}_i}{\partial q} + \frac{\partial \delta q^\star}{\partial \phi_i} \frac{\partial \phi_i}{\partial q}, \tag{30a}$$

$$\frac{\partial \delta q^\star}{\partial u} = \frac{\partial \delta q^\star}{\partial b} \frac{\partial b}{\partial u}. \tag{30b}$$

Similar to other differentiable simulators based on implicit time-stepping [23], [55], we compute the derivatives of the solution $\delta q^\star$ with respect to the problem data $(\mathbf{Q}, b, \mathbf{J}_i, \phi_i)$ by applying the implicit function theorem to the KKT optimality conditions of the convex program (27) [29]. Then, the derivatives of $(\mathbf{Q}, b, \mathbf{J}_i, \phi_i)$ with respect to q and u can be straightforwardly computed using either automatic differentiation or a more specialized method that takes advantage of the structure of rigid body systems [56].

As the derivatives (30) are discontinuous functions of (q, u) [16], they are not a good local approximation of the CQDC dynamics unless multiple gradients are computed for different state-action pairs and averaged using the first-order randomized smoothing scheme (Sec. II-C2).

D. Implementation

As all existing rigid-body differentiable simulators, to the best of our knowledge, assume second-order Newtonian dynamics, it is difficult and inefficient to modify them to support our CQDC dynamics formulation. Therefore, we implemented the proposed dynamics formulation using the Drake robotics toolbox [32]. Although our implementation is not heavily optimized, it is adequate for computing contact-rich plans within a reasonable amount of wall-clock time (a few minutes).

For the forward dynamics (Sec. III-B), we use Drake's MultibodyPlant and SceneGraph for collision detection and the computation of the object mass matrix $\mathbf{M}_\mathrm{u}$, contact Jacobians $\mathbf{J}_i$ and signed distances ϕ_i. The SOCP (27) can then be constructed with MathematicalProgram and solved with a third-party solver of our choice.

For the dynamics derivatives (Sec.III-C), we have a custom implementation for differentiating through the KKT optimality conditions of an SOCP using Eigen's [57] linear solvers, allowing us to efficiently compute the partial derivatives of $\delta q^\star$ with respect to $(\mathbf{Q}, b, \mathbf{J}_i, \phi_i)$ from primal-dual solutions $(\delta q^\star, \lambda^\star)$ given by third-party conic solvers. As for the partials of $(\mathbf{Q}, b, \mathbf{J}_i, \phi_i)$ with respect to q, we note that b is a linear function of q; $\partial \phi_i / \partial q = \mathbf{J}_{\mathrm{n}_i}$; $\partial \mathbf{J}_i / \partial q$ and $\partial \mathbf{Q} / \partial q$ are computed with Drake's forward-mode automatic differentiation.

Finally, in order to avoid discontinuities coming from collision detection, we curate our system models so that every contact pair is either sphere-sphere, sphere-box, or sphere-cylinder, which means the contact points and normals change smoothly with the system configuration q [58]. For example, both the box-shaped fingers of the Allegro hand in Fig. 1a-d and the box-shaped manipuland in Fig. 1b are represented as arrays of inscribing spheres for the purpose of collision detection. We note that this limitation can be alleviated with smoothing over collision geometries [59], [60].

IV. Smoothing of Contact Dynamics

In this section, we combine our formalism and the computation methods of smoothing in Sec.II and the CQDC dynamics in Sec.III, resulting in a class of methods that efficiently make local approximations to the smooth surrogate of our convex quasi-dynamic contact model. Throughout this section, we assume we compute a local model of the smooth surrogate around some nominal state-input pair $(\bar{q}, \bar{u})$.

A. Randomized Smoothing of Contact Dynamics

We follow the method in Sec.II-D in order to perform randomized smoothing of the contact dynamics. For first-order randomized smoothing, we utilize (21) with access to gradients of the contact dynamics obtained in Sec.III-C. We similarly do zero-order randomized smoothing using (22).

However, there is a caveat in the randomized smoothing scheme: if the sampling distribution ρ has infinite support, the sampled state $\bar{q} + w_i^q$ could violate the non-penetration constraint for rigid-bodies, i.e. $\phi(\bar{q} + w_i^q) < 0$ for some i.

Although reasoning about the dynamics $f(q, u)$ with an infeasible (penetrating) q may seem ill-posed [61], we can *define* the dynamics from an infeasible q as the projection of q to the "nearest" point in the feasible (non-penetrating) set. The notion of nearest can be defined in terms of the work required to move the system configuration by δq, which is precisely the quadratic cost (27a) (divided by the step size h). This projection problem can then be written as

$$\min_{\delta q} \; \frac{1}{2}\delta q^\intercal \mathbf{Q} \delta q + b^\intercal \delta q, \text{ subject to} \tag{31a}$$

$$\phi_i(q + \delta q) \geq 0, \; i \in \{1 \ldots n_c\}, \tag{31b}$$

where (31b) is the non-linear non-penetration constraint.

While the projection in (31) is difficult to solve in general, we can linearize the constraint (31b) in order to locally approximate the problem as a QP. When the constraint is linearized, the problem remarkably becomes equivalent to the frictionless special case of the CQDC dynamics (27). In other words, projection is simply another interpretation of what the CQDC dynamics does within the penetrating regime, and no other explicit treatment is required for projection other than the evaluation of CQDC dynamics. In practice, due to the local nature of linearization, we use a low-variance distribution ρ to sample states, though higher variance can be used for inputs.

When samples within the penetrating regime are projected onto the boundary of the feasible set and then averaged, the expected value of such a distribution creates a *stochastic force field* that pushes the object away from feasible set's boundary. We illustrate this phenomenon through a simple example.

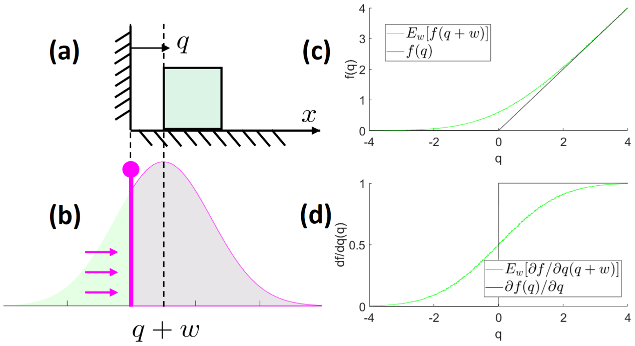


Fig. 4: Figure for Example 3, where quasi-dynamics of motion is interpreted as a projection operator. (**a**) Illustration of the system. (**b**) Distribution of $q+w$ (green) and $f(q+w)$ (pink). Note that the samples for which $q + w_i < 0$ have been projected onto the surface into a delta function, and the expectation of the pink distribution lies on the right side of q, creating a stochastic force field effect. (**c**) CQDC dynamics and its randomized smoothed version. (**d**) Gradients of the CQDC dynamics obtained with first-order randomized smoothing.

Example 3. (The Stochastic Force Field) Consider the dynamics of an unactuated 1D block with a wall occupying $q \leq 0$ (Fig. 4a), such that the physical dynamics is identity *if* the block is in a non-penetrating configuration, $f(q) = q$ if $q \geq 0$. The dynamics within the penetrating regime is not well-defined physically; yet, applying the quasi-dynamic equations of motions (31) to this system gives

$$\min_{\delta q} \; \frac{1}{2}m(\delta q)^2, \text{ subject to} \tag{32a}$$

$$q + \delta q \geq 0. \tag{32b}$$

which has the following solution,

$$f(q) = q + \delta q = \begin{cases} q & \text{if } q \geq 0 \text{ (no penetration)} \\ 0 & \text{else (penetration)} \end{cases} \tag{33}$$

One interpretation of (33) is that configurations inside the wall gets projected onto the wall. By taking an average according to such dynamics, the expectation pushes the box away from the wall as illustrated in Fig. 4, which creates a stochastic force field. Note that with this interpretation, the gradients are also well-defined within the penetrating regime - an infinitesimal change of position in the penetrating configuration does not have any effect on the location of projection in this example, thus $\partial f(q)/\partial q = 0$ if $q < 0$ (But note that after smoothing, $\partial f_\rho(q)/\partial q > 0$). For more complex geometries, the location of projection changes due to changes in the surface, which we connect back to the presence of the $\partial \mathbf{J}/\partial q$ and $\partial \phi/\partial q$ terms in (30a).

B. Analytic Smoothing of Contact Dynamics

Randomized smoothing simply involves solving (27) repeatedly and averaging the resulting q_+, $\mathbf{A}$, and $\mathbf{B}$. In contrast, analytic smoothing (Sec.II-C1) of the CQDC dynamics (27) is not as straightforward, since the solution to an optimization problem does not give easy access to explicit forms from which we can analytically design smooth surrogates.

To perform analytic smoothing, we can convert the hard constraints in a convex program into costs using a penalty method. Specifically, we convert the constrained program (27) into an unconstrained convex program using the log-barrier function, a common technique in the interior-point method for convex conic programs, with weight κ:

$$\begin{aligned}\min_{\delta q}.\ &\frac{1}{2}\delta q^\intercal \mathbf{Q}\delta q + b^\intercal \delta q \\ &- \frac{1}{\kappa}\sum_{i=1}^{n_\mathrm{c}}\log\left[\frac{(\mathbf{J}_{\mathrm{n}_i}\delta q+\phi_i)^2}{\mu_i^2} - (\mathbf{J}_{\mathrm{t}_i}\delta q)^\intercal \mathbf{J}_{\mathrm{t}_i}\delta q\right],\end{aligned} \tag{34}$$

whose solution converges to the solution of (27) as $\kappa \to \infty$ [33, §11.3 and §11.6].

The log-barrier term in (34) can also be interpreted as the potential of a force field whose strength is inversely proportional to the distance to the boundary of the constraint [33, p.567]. For moderate values of κ, constraints can exert forces even though they are not active, achieving a smoothing effect similar to the "force-at-a-distance" relaxation of complementarity constraints, which are commonly used in planning through contact methods such as [15], [23]. We will further illustrate this similarity in Example 4.

From $\delta q^\star$, the solution to the smoothed dynamics (34), we can directly compute the smoothed gradient $\mathbf{A}_\rho$ and $\mathbf{B}_\rho$ in the same way as (29). Once again, the derivatives of $\delta q^\star$ with respect to q and u are computed by applying the implicit function theorem to the optimality condition of (34), which only consists of the stationarity condition due to the absence of conic constraints.

To solve (34), we implemented an in-house solver using Newton's method [33, §9.5], and find that our out-of-the-textbook implementation works robustly and reliably for all numerical experiments in Sec. V and VIII.

C. Equivalence of Smoothing Schemes

In Sec.II, we saw that randomized and analytic smoothing can be interpreted as different methods of computing the same quantity. Here, we show with examples that (**i**) for simple systems, we can derive the sampling distribution ρ needed to obtain the smoothing given by the log-barrier-based analytic smoothing scheme; (**ii**) for more complex systems, randomized smoothing schemes using a Gaussian ρ and the log-barrier-based analytic smoothing scheme result in qualitatively-similar smoothed dynamics.

Example 4. (**Equivalence of Smoothing Schemes**) We start with the 1D frictionless system in Fig. 5a, whose dynamics is an instance of the frictionless CQDC dynamics (27):

$$\begin{aligned}\min_{\delta q^\mathrm{a}}.\ &\frac{1}{2}hk_\mathrm{a}(\delta q^\mathrm{a})^2 - hk_\mathrm{a}(u-q^\mathrm{a})\delta q^\mathrm{a},\ \text{subject to} \\ &q^\mathrm{a}+\delta q^\mathrm{a} \geq 0,\end{aligned} \tag{35}$$

The KKT conditions of (35) are also the equations of motion of the system:

$$hk_\mathrm{a}\left(q^\mathrm{a}+\delta q^\mathrm{a}-u\right) = \lambda, \tag{36a}$$

$$0 \leq q^\mathrm{a}+\delta q^\mathrm{a} \perp \lambda \geq 0, \tag{36b}$$

which has the explicit solution

$$q^\mathrm{a}_+ = q^\mathrm{a}+\delta q^\mathrm{a} = \begin{cases} u & \text{if } u \geq 0 \text{ (no contact)} \\ 0 & \text{else (contact)}\end{cases}. \tag{37}$$

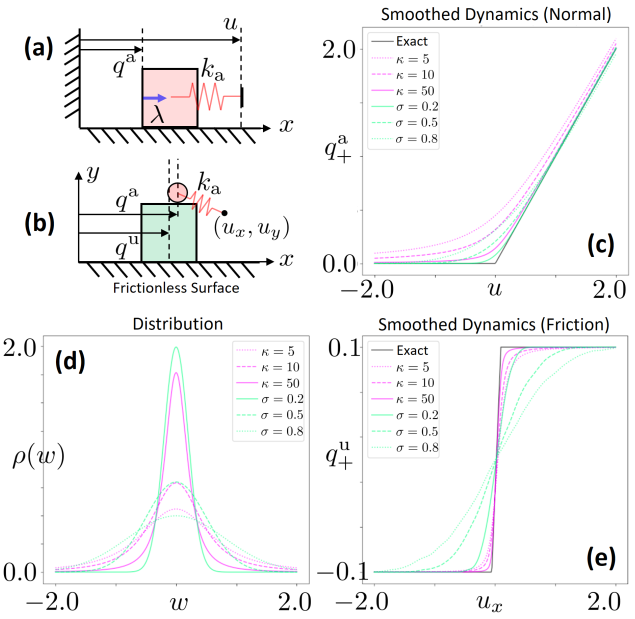


Fig. 5: (**a**) A system consisting of an actuated cart constrained to slide on a frictionless surface, and a wall occupying $q^\mathrm{a} \leq 0$. The actuator has stiffness k_a. (**b**) A system consisting of an unactuated cart constrained to slide on a frictionless surface, and a ball actuated along both the x and y axes. The ball can touch the top surface of the cart with a frictional contact. (**c**) Randomized and analytic smoothing of (a). Randomized smoothing, shown in green, is done with a Gaussian kernel with different variances σ. Analytic smoothing, shown in magenta, is done with different log-barrier weights κ. (**d**) Density functions of the Gaussian kernels (green) and the elliptical distributions used for analytic smoothing (magenta). (**e**) Randomized and analytic smoothing of (b). We plot q^u_+ against u_x for a fixed u_y that is inside the cart. The linear region in the plot corresponds to sticking contact, and the flat regions to sliding.

Also as an instance of (34), (35) can be smoothed analytically by converting the constraint into a cost using the log-barrier function:

$$\min_{\delta q^\mathrm{a}}.\ \frac{1}{2}hk_\mathrm{a}(\delta q^\mathrm{a})^2 - hk_\mathrm{a}(u-q^\mathrm{a})\delta q^\mathrm{a} - \frac{1}{\kappa}\log\left(q^\mathrm{a}+\delta q^\mathrm{a}\right), \tag{38}$$

whose optimality condition is obtained by setting the gradient of the smoothed cost to 0, yielding the equations of motion of the smoothed system:

$$hk_\mathrm{a}(q^\mathrm{a}+\delta q^\mathrm{a}-u) = \frac{1}{\kappa}\left(\frac{1}{q^\mathrm{a}+\delta q^\mathrm{a}}\right). \tag{39}$$

The right-hand side of (39) can be interpreted as an impulse whose magnitude is inversely proportional to the distance to the wall. Calling this impulse β, we note that

$$\beta(q^\mathrm{a}+\delta q^\mathrm{a}) = \kappa^{-1}, \tag{40}$$

which is analogous to the common bilinear relaxation to the complementarity constraint for contact [15], [23].

The solution to (39) is given by

$$q^\mathrm{a}_+ = \frac{1}{2}\left(u + \sqrt{u^2 + \frac{4}{\kappa h^2 k_\mathrm{a}^2}}\right) \tag{41}$$

which is equivalent to randomized smoothing of the original dynamics under the following elliptical distribution

$$\rho(w) = \sqrt{\frac{4\sigma}{(w^\intercal \sigma w + 4)^3}}, \tag{42}$$

where $\sigma := hk\kappa$. We note that as opposed to the Gaussian, this distribution has a heavy tail.

For more complex examples involving friction, such as the system in Fig. 5b, obtaining an analytic expression for ρ that corresponds to (34) is difficult. Instead, we numerically illustrate the performance of the two smoothing schemes in Fig. 5e. (See [16] for more details). We note that with a different relaxation of complementary slackness, Howell *et al.* showed a similar trend for the smoothed dynamics as the relaxation is tightened [23, Fig. 7].

For more general settings beyond our simple examples, we expect there to be subtle mathematical differences between analytic and randomized smoothing, though they demonstrate similar empirical behavior in our work.

V. Case Study: Trajectory Optimization

Throughout the previous sections, we have developed the theory and computational methods to obtain local linear models of the smooth surrogate of the proposed CQDC dynamics. These tools are not specific to any planning algorithm: they can improve the performance of most iterative planning algorithms that rely on gradients.

In this section, we demonstrate the efficacy of smoothed CQDC dynamics on trajectory optimization for systems with contacts. Although smoothing of hard contact constraints has been widely utilized to improve convergence [15], [23], [62], existing methods still struggle with complex problems such as dexterous manipulation. In particular, the quality of solutions can be very sensitive to initial guesses [34]. In this context, we show that a variant of trajectory optimization, with the help of smoothed CQDC dynamics and only a trivial initial guess, can perform well even on dexterous manipulation tasks.

A. Iterative MPC with Smoothing

The variant of trajectory optimization algorithm used in this section, which we call iterative MPC (iMPC), is an iLQR-inspired algorithm proposed in [16]. We briefly summarize iMPC here for completeness, and illustrate how smoothing can be easily integrated into iMPC.

Consider the problem of finding an optimal sequence of inputs to track some desired state trajectory $\{x_t^d\}_{t=0}^{T}$. We need an initial guess for the nominal input trajectory $\{\bar{u}_t\}_{t=0}^{T-1}$, from which the nominal state trajectory $\{\bar{x}_t\}_{t=0}^{T}$ can be obtained by rolling out $\{\bar{u}_t\}_{t=0}^{T-1}$ from the initial state x_0. For every time t (i.e. in a time-varying manner), we can create a locally linear model that approximates the dynamics, with model parameters $\{\mathbf{A}_t, \mathbf{B}_t, c_t\}_{t=0}^{T-1}$ (18). Then, finding the optimal $\{\bar{u}_t\}_{t=0}^{T-1}$, subject to the locally linear model of the dynamics, can be written as a QP. We present the MPC variant of this problem that receives the initial state $\bar{x}_j$ at time $t = j$ and computes the optimal action for the remaining time steps,

$$\mathbf{MPC}(\bar{x}_j) = u_j^\star, \text{where} \tag{43a}$$

$$\min_{x_t, u_t} \quad \left\| x_T - x_T^d \right\|_{\mathbf{Q}_T}^2 + \sum_{t=j}^{T-1} \left(\|x_t - x_t^d\|_{\mathbf{Q}_t}^2 + \|u_t\|_{\mathbf{R}_t}^2 \right) \tag{43b}$$

$$\text{s.t.} \quad x_{t+1} = \mathbf{A}_t(x_t - \bar{x}_t) + \mathbf{B}_t(u_t - \bar{u}_t) + c_t, \tag{43c}$$

$$\mathbf{C}_t^x x_t \leq d_t^x, \; \mathbf{C}_t^u u_t \leq d_t^u, \; \forall t \in \{j \cdots T-1\}, \tag{43d}$$

$$x_j = \bar{x}_j. \tag{43e}$$

Here, $\{\mathbf{Q}_t, \mathbf{R}_t\}$ are the quadratic weights for state and input, respectively; $\mathbf{Q}_T$ is the weight on the terminal state; $\{\mathbf{C}_t^x, d_t^x\}$ and $\{\mathbf{C}_t^u, d_t^u\}$ are inequality parameters on the state and input, respectively. The linear constraints (43d) can enforce, for instance, joint and actuation limits.

The iMPC algorithm is summarized in Alg. 1. In every outer iteration (body of the `while` loop starting at Line 4), iMPC solves truncated versions of (43) for $T-1$ times. Specifically, at inner iteration j (body of the `for` loop starting at Line 6), we solve MPC (43) for the sub-problem starting at $t = j$ (Line 7), and apply $u_j^\star$ from the solution to update $\bar{x}_{j+1}$ (Line 8). During MPC, we also enforce a trust region by using (43d) to constrain x_t and u_t to stay close to $\bar{x}_t$ and $\bar{u}_t$, respectively.

Algorithm 1: iMPC

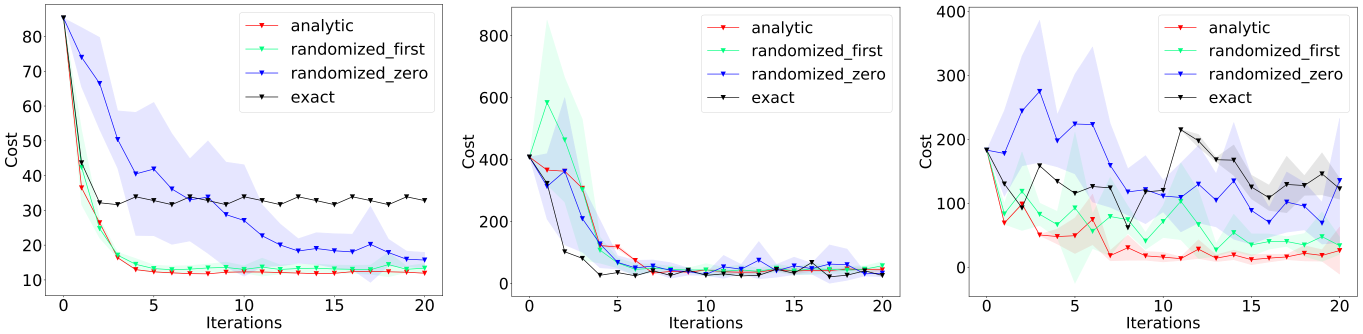
```
1 Input: Initial state $x_0$, input trajectory guess $\{\bar{u}_t\}_{t=0}^{T-1}$;
2 Output: Optimized input trajectory $\{\bar{u}_t\}_{t=0}^{T-1}$ ;
3 $\{\bar{x}_t\}_{t=0}^{T}$ ← Rollout $f$ from $x_0$ with$\{\bar{u}_t\}_{t=0}^{T-1}$;
4 while not converged do
5     Compute system matrices $\{\mathbf{A}_t, \mathbf{B}_t, c_t\}_{t=0}^{T-1}$;
6     for $0 \leq j < T$ do
7         $\bar{u}_j \leftarrow \mathbf{MPC}(\bar{x}_j)$ ;
8         $\bar{x}_{j+1} \leftarrow f(\bar{x}_j, \bar{u}_j)$ ;
9 return $\{\bar{u}_t\}_{t=0}^{T-1}$
```

To apply smoothing to iMPC, we substitute the linearizations of smooth surrogates $\{\mathbf{A}_{t,\rho}, \mathbf{B}_{t,\rho}, c_{t,\rho}\}_{t=0}^{T-1}$ (20) for the first-order Taylor expansions $\{\mathbf{A}_t, \mathbf{B}_t, c_t\}_{t=0}^{T-1}$ (18). After every outer iteration, we reduce the variance of ρ, allowing the smooth surrogates f_ρ to converge to true dynamics f. For analytic smoothing, we initially set κ using physical intuition from (40), and increase it every iteration.

B. Experiment Setup

1) Systems Description : We test iMPC with different smoothing schemes on two planar systems from [16] and a 3D system for in-hand rotation. We describe our systems below, and their visualization can be seen in Fig. 11. The three tuple after the name of each system indicates $(n_\mathrm{u}, n_\mathrm{a}, n_\mathrm{cg})$, where n_u is the number of unactuated DOFs, n_a the number of actuated DOFs, and n_cg the number of collision geometries.

1) **Planar Pushing, (3,2,2)**. A classical example of non-prehensile manipulation [63]. The goal is specified as some 2D configuration of the box.
2) **Planar Hand Reorientation, (3,4,13)**. We use a planar hand with two fingers, each with two DOFs. The goal is to change the position and orientation of the ball in a 2D plane.



(a) PlanarPushing.
(b) PlanarHand Re-orientation.
(c) AllegroHand Rotation.

Fig. 6: Performance of iMPC with different smoothing schemes: analytic, randomized (first-order), randomized zero-order, and exact (no smoothing). For each method, the solid line represents the mean over five runs, and the shaded region represents the standard deviation.

3) **Allegro In-Hand Rotation, (6,16,20)**. 3D In-hand rotation of the ball with the full model of the Allegro hand [14]. The goal is specified as a rotated configuration of the ball.

2) Initialization: As mentioned in Sec. V-A, given an initial state x_0, we need to initialize the nominal input trajectory $\{\bar{u}_t\}_{t=0}^{T-1}$, where $\bar{u}_t$ is the commanded positions of the robots at step t under the CQDC dynamics. Empirically, we find that good convergence can be achieved with a constant initialization, i.e. $\bar{u}_0 = \bar{u}_1 = \dots \bar{u}_{T-1}$.

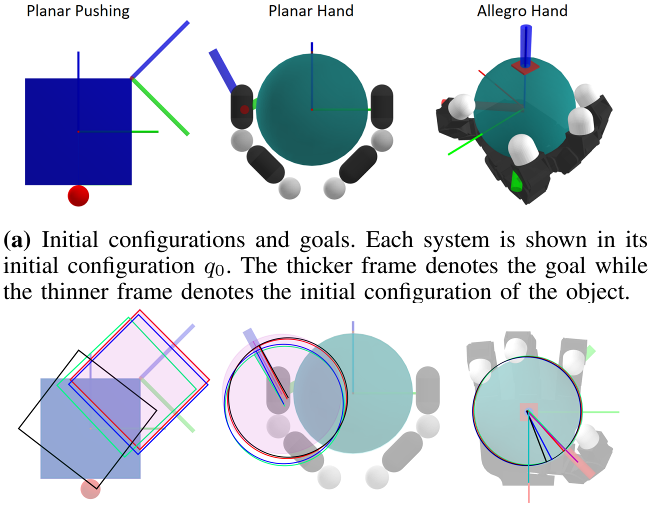
For the numerical experiments in this section, we need a $\bar{u}_0$ that makes contact with the object. Otherwise the baseline which does not use smoothing would have zero gradients and make no progress at all.

In contrast, for iMPC with smoothing, it is sufficient to set $\bar{u}_0 = q_0^{\mathrm{a}}$, as long as q_0^{a} is not “too far away” from making contact with the objects (the reason is explained in Example 6). In many practical problems, the object's initial configuration q_0^{u} is fixed, but we are free to choose the initial robot configuration q_0^{a}. In this case, we can simply calculate a q_0^{a} that is “close” to making contact using, for example, methods that compute grasps [64].

3) Hardware & Implementation Details: The numerical experiments are run on a desktop with one AMD Threadripper 2950 CPU (16 cores, 32 threads) and 32GB of RAM. The code for iMPC using different smoothing schemes is identical except for the computation of the linearizations. For analytic smoothing and the baseline, we solve respectively the smoothed (34) and original (27) dynamics once and then apply the chain rule to get the linearization. For first-order randomized smoothing, we solve the original dynamics (27) and apply the chain rule for 100 samples ($N = 100$), which is parallelized on all available threads, and then average the gradients of the samples. Zeroth-order randomized smoothing simply requires parallel evaluation of the dynamics.

C. Results & Discussion

In Fig. 6, we plot the performance of iMPC with a baseline that does not use smoothing, and the three different smoothing schemes in Sec. II-C, namely analytic, randomized first-order, and randomized zeroth-order. We also summarize the running time of each method, as well as the minimum cost achieved across the iterations, in Table II. Illustrations of the tasks and the results achieved by different methods are shown in Fig. 7. We interpret the results and discuss the relevant findings in this section.



(a) Initial configurations and goals. Each system is shown in its initial configuration q_0. The thicker frame denotes the goal while the thinner frame denotes the initial configuration of the object.

(b) Final object configuration achieved by the best runs within each of the four methods. Pink shaded denotes the goal configuration for the first two examples, while the goal configuration in the last example is marked by the pink line protruding out of the object. Colors correspond to the plots in Fig. 6.

Fig. 7: Tasks and results for the trajectory optimization case study.

Problem	PlanarPushing		PlanarHand		AllegroHand	
Method	Cost	Time(s)	Cost	Time(s)	Cost	Time(s)
A.	11.74	2.17	26.55	5.20	**5.78**	19.59
RF.	**11.73**	4.64	**17.87**	23.09	8.42	40.07
RZ.	12.86	4.61	18.29	11.93	28.21	34.05
Exact	31.64	1.88	18.49	5.91	44.68	12.92

TABLE II: Average of minimum cost and running time achieved by different methods. All methods are ran for 20 iterations across 5 trials. Method abbreviations: A(Analytic), RF(Randomized First), RZ(Randomized Zero).

1) Exact vs. Smoothing: For PlanarPushing and AllegroRotation, the various smoothing schemes achieve much lower costs than using exact gradients. However, for PlanarHand, using the exact linearization is performant as well. This difference may be explained by the observation that the planar hand example does not go through many mode changes, while the planar pusher and the Allegro hand require several mode changes to converge to the optimal trajectory.

2) Analytic vs. Randomized Smoothing: Comparing the performance of the three smoothing schemes, the analytic and the first-order randomized smoothing perform similarly, while the zeroth-order version does not perform as well. We believe the cause lies in the high variance characteristic of the zeroth-order estimator in higher dimensions.

3) Running (wall-clock) Time: While analytic smoothing only requires one evaluation of the smoothed dynamics (34) in order to compute $(\mathbf{A}_\rho, \mathbf{B}_\rho, c_\rho)$, randomized smoothing requires taking N samples and averaging them, which costs N times more compute-time. After parallelization, we expect randomized smoothing to be roughly N/ξ times slower than analytic smoothing where ξ is the number of threads. Indeed, with $N = 100$ and $\xi = 32$, our results show that randomized smoothing is 2 to 3 times slower than analytic smoothing.

VI. Local Mahalanobis Metric for RRT

While trajectory optimization can find trajectories reaching goals that are close to the initial configuration, it is highly prone to local minima when the goal is further away (e.g. moving the box back in PlanarPushing, rotating the ball by 180 degrees in PlanarHand, AllegroHand). To solve these tasks, the planning algorithm needs to be more global. When faced with such problems, the RRT algorithm [35] has proven to be a classical and effective method for global planning.

However, extending RRT to dynamical systems (i.e. kinodynamic RRT) has been difficult, as a distance metric between two states is hard to define. In [6], it was argued that a good distance metric for RRT would need to explicitly consider dynamic reachability in order to efficiently grow the tree. The authors further proposed Reachability-Guided RRT (RG-RRT), which had system-specific reachability metrics that was shown to be effective for smooth systems. To alleviate the limitation of being system-specific, later works have considered building such metrics based on local characteristics of the dynamics such as local linearizations [7], [8].

However, when the dynamics involves contact, such local linearizations are no longer informative, and existing approaches often tackle dynamic reachability by explicitly considering contact modes [9], [10]. This has led to planners that scale poorly with the number of contacts. In contrast, we propose to handle the challenges brought about by contact with smoothing. We show that when combined with smoothing, the locally linear model can be used to construct an informative distance metric that is consistent with notions of reachability.

A. The Local Mahalanobis Metric

Consider the following problem: given the current configuration $\bar{q}$, and some queried configuration q, how can we formulate a distance metric $d(q; \bar{q})$ that is consistent with reachability characteristics of the system? We propose to utilize the locally linear model around the nominal configuration $\bar{q}$, that characterizes the local response of the next system configuration q_+ with respect to the movement of the actuated configurations u. This local model can be written as

$$q_+ = \mathbf{B}(\bar{q}, \bar{q}^{\mathrm{a}}) \underbrace{(u - \bar{q}^{\mathrm{a}})}_{\delta u} + c(\bar{q}, \bar{q}^{\mathrm{a}}) \tag{44}$$

where the notation is consistent with the CQDC dynamics formulation in Sec.III-B; the input u is the position command to the system, and $\mathbf{B}, c$ are defined as (18). We note that the contribution of $\mathbf{A}$ term is zero since $\delta x = 0$. Also note that we set the nominal action to be $\bar{u} = \bar{q}^{\mathrm{a}}$.

Given such a local characterization, and a queried state q, we define the Mahalanobis metric as follows.

Definition 4. Local Mahalanobis Metric. Given a nominal configuration $\bar{q}$, and queried configuration q, we define the Mahalanobis distance d_γ of q from $\bar{q}$ as follows:

$$\begin{aligned} d_\gamma(q; \bar{q}) &:= \|q - \mu\|_{\mathbf{\Sigma}_\gamma^{-1}} = \tfrac{1}{2}(q - \mu)^\intercal \mathbf{\Sigma}_\gamma^{-1}(q - \mu) \\ \mathbf{\Sigma}_\gamma &:= \mathbf{B}(\bar{q}, \bar{q}^{\mathrm{a}})\mathbf{B}(\bar{q}, \bar{q}^{\mathrm{a}})^\intercal + \gamma \mathbf{I}_n, \ \mu := c(\bar{q}, \bar{q}^{\mathrm{a}}). \end{aligned} \tag{45}$$

The regularization $\gamma\mathbf{I}_n$ is added to ensure that $\mathbf{\Sigma}_\gamma$ is positive definite and the inverse $\mathbf{\Sigma}_\gamma^{-1}$ is well-defined. Note that the ε-sublevel set of this metric $d_\gamma(q; \bar{q})$, which we denote by $\mathcal{R}_{\varepsilon,\gamma}(\bar{q})$, describes an *ellipsoid* that is centered at μ and has a shape matrix $\mathbf{\Sigma}_\gamma^{-1}$. We further motivate our construction of the metric by noting that this ellipsoid can be alternatively characterized (equivalent up to regularization) by the set

$$\mathcal{R}_\varepsilon(\bar{q}) := \{\mathbf{B}(\bar{q}, \bar{q}^{\mathrm{a}})\delta u + c(\bar{q}, \bar{q}^{\mathrm{a}}) \mid \|\delta u\| \leq \varepsilon\}. \tag{46}$$

This equivalence relation is exact when $\mathbf{B}$ has full row rank (i.e. the system is one-step controllable) and $\gamma = 0$. On the other hand, if $\mathbf{B}$ loses rank, one of the principle axis of $\mathcal{R}_\varepsilon$ has a length of zero and the set becomes degenerate.

Finally, note that when $q - \mu \notin \mathbf{Range}(\mathbf{B})$, i.e. there is no actuation u that can take the state $\bar{q}$ to the queried state q, the distance $d_\gamma(q; \bar{q})$ is a large number dominated by the inverse of the regularization term γ^{-1}, which is consistent with the intuition that states that are harder to reach are further away.

B. Metric on Smoothed Dynamics and Unactuated Objects

As explained in the previous sections, the local model constructed using $\mathbf{B}$ may not be a very informative one for non-smooth systems with contact. In light of the various smoothing schemes introduced in Sec.II to alleviate this issue, we propose a metric by utilizing the linearization of the smooth surrogate $(\mathbf{B}_\rho, c_\rho)$ as described in (20), as opposed to those of the original contact dynamics, $(\mathbf{B}, c)$.

Furthermore, for systems where robots interact with unactuated objects through contact, we focus on the reachability of the objects, as the robots are actuated and can easily move to a desired configuration without contact. We combine smoothing and the object-centric reachability in the following variant of the Mahalanobis metric $d^{\mathrm{u}}_{\rho,\gamma}$,

$$\begin{aligned} d^{\mathrm{u}}_{\rho,\gamma}(q; \bar{q}) &:= \|q^{\mathrm{u}} - \mu^{\mathrm{u}}_\rho\|_{\mathbf{\Sigma}^{\mathrm{u}\,-1}_{\rho,\gamma}}, \\ \mathbf{\Sigma}^{\mathrm{u}}_{\rho,\gamma} &:= \mathbf{B}^{\mathrm{u}}_\rho(\bar{q}, \bar{q}^{\mathrm{a}})\mathbf{B}^{\mathrm{u}}_\rho(\bar{q}, \bar{q}^{\mathrm{a}})^\intercal + \gamma \mathbf{I}_{n_{\mathrm{u}}}, \\ \mu^{\mathrm{u}}_\rho &:= c^{\mathrm{u}}_\rho(\bar{q}, \bar{q}^{\mathrm{a}}). \end{aligned} \tag{47}$$

where $\mathbf{B}^{\mathrm{u}}_\rho$ is formed by the rows of $\mathbf{B}_\rho$ corresponding to the unactuated DOFs, and c^{u}_ρ is defined similarly. Finally, we define $\mathcal{R}^{\mathrm{u}}_{\rho,\varepsilon,\gamma}(\bar{q})$ as the ε-sublevel set of $d^{\mathrm{u}}_{\rho,\gamma}(q; \bar{q})$.

In the rest of this section, we give several examples that provide intuition into the local Mahalanobis metric $d^{\mathrm{u}}_{\rho,\gamma}$ and its sublevel set $\mathcal{R}^{\mathrm{u}}_{\rho,\varepsilon,\gamma}$.

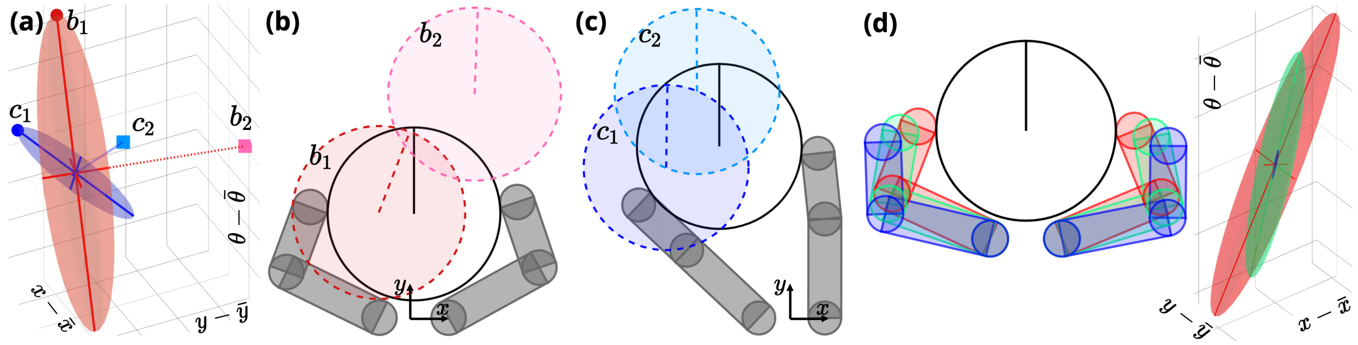


Fig. 8: (**a**) Two different sublevel sets $\mathcal{R}^{\mathrm{u}}_{\rho,\varepsilon,\gamma}$, represented as ellipsoids, shown in the space of q^{u}, with $\varepsilon = 1$, and $\gamma = 10^{-6}$. The ellipsoid centers are shifted to the origin for easy comparison. Red ellipsoid: $\mathcal{R}^{\mathrm{u}}_{\rho,\varepsilon,\gamma}$ for the system configuration in Fig. 8b; blue ellipsoid: $\mathcal{R}^{\mathrm{u}}_{\rho,\varepsilon,\gamma}$ for the configuration in Fig. 8c. Points b_1, c_1 are where ellipsoids' major axes intersect their boundaries. Points b_2, c_2 are points along the minor axes of the ellipsoids, and satisfy $\|b_1\| = \|b_2\|$ and $\|c_1\| = \|c_2\|$, where the norm is based on the standard Euclidean metric. (**b**) The solid robots and objects represent the $\bar{q}$ at which the red $\mathcal{R}^{\mathrm{u}}_{\rho,\varepsilon,\gamma}$ in (a) is computed. The straight line on the puck indicates its orientation. The dashed dark red puck corresponds to the configuration b_1, and pink to b_2. Note that b_1 is easier to each than b_2. (**c**) Similar to Fig. 8b, the dashed dark blue puck correspond to c_1, and light blue to c_2. It is also easier to reach c_1 than c_2. (**d**) The volume of $\mathcal{R}^{\mathrm{u}}_{\rho,\varepsilon,\gamma}$ shrinks as the fingers get further away from the puck. The ellipsoids on the right are color-coded to match the robot configurations on the left. Note that the blue ellipsoid is barely visible.

Example 5. (Understanding $\mathbf{B}$ for Planar Systems) When both the robots and objects are constrained in a plane (e.g. the planar systems in Sec. V-B1), the CQDC dynamics (27) simplifies to the following QP:

$$\min_{\delta q}. \ \frac{1}{2}\delta q^{\intercal}\mathbf{Q}\delta q + b^{\intercal}\delta q, \text{ subject to} \tag{48a}$$

$$(\mathbf{J}_{\mathrm{n}_i} + \mu_i \mathbf{J}_{\mathrm{t}_i})\delta q + \phi_i \geq 0, \ i \in \{1 \ldots n_{\mathrm{c}}\}, \tag{48b}$$

$$(\mathbf{J}_{\mathrm{n}_i} - \mu_i \mathbf{J}_{\mathrm{t}_i})\delta q + \phi_i \geq 0, \ i \in \{1 \ldots n_{\mathrm{c}}\}, \tag{48c}$$

where the contact Jacobian $\mathbf{J}_{\mathrm{t}_i}$ has only one row instead of two, and the conic contact constraint (27b) reduces to two inequality constraints (48b) and (48c). We define $\mathbf{J} \in \mathbb{R}^{(2n_{\mathrm{c}}) \times n_q}$ by stacking the $\mathbf{J}_{\mathrm{n}_i} + \mu_i \mathbf{J}_{\mathrm{t}_i}$ and $\mathbf{J}_{\mathrm{n}_i} - \mu_i \mathbf{J}_{\mathrm{t}_i}$ from (48b) and (48c) into a single matrix, and partition $\mathbf{J}$ into $\mathbf{J}_{\mathrm{u}}$ and $\mathbf{J}_{\mathrm{a}}$ in a similar way as in (24).

More structure behind the $\mathbf{B}$ matrix (as defined in (29)) can be revealed with a bit of linear algebra. We can work out by hand the application of the implicit function theorem to the KKT conditions of (48), and the chain rule in (30b), to obtain an explicit expression for $\mathbf{B}$:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}^{\mathrm{a}} \\ \mathbf{B}^{\mathrm{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{I} - (h^2\mathbf{K}_{\mathrm{a}})^{-1}(\tilde{\mathbf{J}}_{\mathrm{a}})^{\intercal}\mathbf{P}\tilde{\mathbf{J}}_{\mathrm{a}} \\ \mathbf{M}_{\mathrm{u}}^{-1}(\tilde{\mathbf{J}}_{\mathrm{u}})^{\intercal}\mathbf{P}\tilde{\mathbf{J}}_{\mathrm{a}} \end{bmatrix}, \text{ with} \tag{49a}$$

$$\mathbf{P} = \left[\tilde{\mathbf{J}}_{\mathrm{u}}\mathbf{M}_{\mathrm{u}}^{-1}(\tilde{\mathbf{J}}_{\mathrm{u}})^{\intercal} + \tilde{\mathbf{J}}_{\mathrm{a}}(h^2\mathbf{K}_{\mathrm{a}})^{-1}(\tilde{\mathbf{J}}_{\mathrm{a}})^{\intercal}\right]^{-1}. \tag{49b}$$

where we assume $\tilde{\mathbf{J}}_{\mathrm{u}}$ and $\tilde{\mathbf{J}}_{\mathrm{a}}$ have full row rank. The tilde over a Jacobian indicates the sub-matrix formed by rows of the original matrix corresponding to the active constraints, i.e. contacts with non-zero contact forces.

The structure in $\mathbf{B}^{\mathrm{u}}$ explains why $\mathbf{B}^{\mathrm{u}}_{\rho}$ is a good measure of the object's reachability when there is contact. We can interpret $\mathbf{Range}(\tilde{\mathbf{J}}_{\mathrm{u}}^{\intercal})$ as achievable object motions under the specific subset of active contacts. By averaging $\mathbf{B}^{\mathrm{u}}$ computed from different contacts which can be activated from the nominal $(\bar{q}, \bar{u})$, $\mathbf{B}^{\mathrm{u}}_{\rho}$ summarizes possible object motions due to contact, in the form of $\mathbf{Range}(\tilde{\mathbf{J}}_{\mathrm{u}}^{\intercal})$ weighted by $\mathbf{M}_{\mathrm{u}}$ and $\mathbf{P}$.

Furthermore, for a configuration with no active contacts, (49) implies that $\mathbf{B}^{\mathrm{a}} = \mathbf{I}$ and $\mathbf{B}^{\mathrm{u}} = \mathbf{0}$, as both $\tilde{\mathbf{J}}_{\mathrm{a}}$ and $\tilde{\mathbf{J}}_{\mathrm{u}}$ are empty matrices in the absence of active contacts. This has the intuitive interpretation that under a u that does not lead to contacts, the robot will move to where it is commanded to, and the object will remain still.

As $\mathbf{B}^{\mathrm{u}}_{\rho}$ is the expected value of $\mathbf{B}^{\mathrm{u}}$, it follows naturally from the above observation that the local distance metric $d^{\mathrm{u}}_{\rho,\gamma}$ tends to be dominated by the regularization $\gamma\mathbf{I}_{n_{\mathrm{u}}}$ for a nominal configuration $\bar{q}$ where robots and objects are far from making contact. In such cases, the probability that an action u sampled from a distribution ρ centered at $\bar{q}^{\mathrm{a}}$ leads to active contacts is low. As a result, in the Monte-Carlo estimation of $\mathbf{B}^{\mathrm{u}}_{\rho}$, such samples simply introduce $\mathbf{0}$ into the average, dragging the distance metric $d^{\mathrm{u}}_{\rho,\gamma}$ towards being dominated by the regularization.

Example 6. (Metric on Planar Hand) We illustrate how the Mahalanobis metric can guide planning using the PlanarHand system first introduced in Sec. V-B. As shown in Fig. 8, the system lives in the xy plane, with gravity pointing into the paper along the negative z direction. The system consists of two actuated 2-link robotic fingers and an unactuated puck which is free to translate and rotate. Each finger can interact with the ball through frictional contacts along both links.

For a given q^{u}, the difficulty of reaching q^{u} from $(\bar{q}, \bar{u})$ can be measured by the local Mahalanobis metric $d^{\mathrm{u}}_{\rho,\gamma}$, whose 1-sublevel sets are shown in Fig. 8a as ellipsoids. Although object configurations b_1 and b_2 are equidistant to the origin under the globally-uniform Euclidean metric, b_1 is considered much closer than b_2 under the local Mahalanobis metric (red ellipsoid). Indeed, in Fig. 8b, reaching b_1 from the current puck configuration seems easier than reaching b_2. A similar observation can be made for the configuration in Fig. 8c.

In addition, the local Mahalanobis metric also varies greatly from one configuration another, as evidenced by the difference between the blue and red ellipsoids in Fig. 8a. This implies that a globally-uniform metric is rarely a good measure of reachability characteristics.

Lastly, the ellipsoid that corresponds to the 1-sublevel set shrinks as the nominal state gets further away from the contact manifold, as shown in Fig. 8d. This signifies that the configurations where the object is less accessible by the robot are naturally considered “further away” and can thus be avoided by the planner.

VII. RRT through Contact

We are now ready to present our smoothing-based enhancements to the vanilla RRT algorithm, which we reproduce in Alg. 2 to establish notations for our discussion. Our method enhances RRT by incorporating (**i**) a reachability-aware `Nearest` operation based on the smoothed Mahalanobis metric on the unactuated objects $d^{\mathrm{u}}_{\rho,\gamma}$, (**ii**) a fast `Extend` operation based on the projection of the subgoal to the range of $\mathbf{B}_\rho$; and (**iii**) a contact sampling procedure which improves the reachability of nodes added to the tree.

We denote the RRT tree as $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$. Each node $q \in \mathcal{V}$ is simply a point in the configuration space of the system.

Algorithm 2: RRT

```
Input: $q_{\mathrm{init}}, q_{\mathrm{goal}}, K$;
Output: $\mathcal{T}$;
$\mathcal{T} = \{q_{\mathrm{init}}\}$;
for $k = 1, \ldots, K$ do
    $q_{\mathrm{subgoal}}$ = SampleSubgoal(p);
    $q_{\mathrm{nearest}}$ = Nearest($q_{\mathrm{subgoal}}$);
    $q_{\mathrm{new}}$ = Extend($q_{\mathrm{nearest}}, q_{\mathrm{subgoal}}$);
    AddNode($q_{\mathrm{new}}$) ;
    if GoalReached then
        break;
```

A. Nearest Node using Local Mahalanobis Metric

As illustrated in Sec.VI, in particular by Example 6, a globally-uniform metric used by the vanilla RRT is usually a poor measure of reachability. Given a subgoal q_{subgoal}, if the nearest node q_{nearest} is chosen under a globally-uniform metric, reaching q_{subgoal} from q_{nearest} may require large u or even be dynamically infeasible. This will compromise RRT’s ability to explore the configuration space, as trying to `Extend` towards a hard-to-reach q_{subgoal} typically returns a child node that is close to the parent node q_{nearest}. In order to retain RRT’s ability to efficiently explore under dynamics constraints, we use the smoothed Mahalanobis metric (47) instead of the usual Euclidean metric in the `Nearest` step:

$$q_{\mathrm{nearest}} = \mathrm{argmin}_{q \in \mathcal{V}}\, d^{\mathrm{u}}_{\rho,\gamma}(q_{\mathrm{subgoal}}; q). \tag{50}$$

B. Dynamically Consistent Extension

After choosing q_{nearest} from the tree $\mathcal{T}$, we need an action or a sequence of actions that moves the system from q_{nearest} to q_{subgoal} subject to the dynamics constraint. One feasible strategy to connect q_{nearest} to q_{subgoal} is to solve for an input sequence $\{u_t\}_{t=0}^{T-1}$ using a trajectory optimization algorithm such as Alg. 1 [36]. However, the high computational cost of trajectory optimization motivates us to seek a simpler solution.

Fortunately, as a result of the farsightedness of quasi-dynamic models, even an input sequence with $T = 1$ (i.e. a single time step) can steer the system fairly far away from q_{nearest}. Although trajectory optimization with $T > 1$ can explore a larger region around q_{nearest}, we find in practice that a single time step is sufficient for `Extend` to effectively grow the RRT tree $\mathcal{T}$.

We present the modified `Extend` that uses a single time step in Alg. 3. The input u is computed by projecting $(q^{\mathrm{u}}_{\mathrm{subgoal}} - \mu^{\mathrm{u}}_\rho)$ to $\mathbf{Range}(\mathbf{B}^{\mathrm{u}}_\rho)$ (20), (47) using least-squares (Line 3). Afterwards, we normalize the input and multiply it by some stepsize ε. The scaled input is then passed to the forward dynamics to obtain a new node. Crucially, we use the actual dynamics f as opposed to the smooth surrogate dynamics f_ρ (Line 4). This ensures that while the search for the next action relies on the smoothed model, the actual path is dynamically consistent under the original non-smooth contact dynamics (i.e. CQDC).

Algorithm 3: `Extend`

```
1 Input: $q_{\mathrm{nearest}}, q_{\mathrm{subgoal}}$;
2 Output: $q_{\mathrm{new}}$;
3 $\delta u^\star = \mathrm{argmin}_{\delta u}\|\mathbf{B}^{\mathrm{u}}_\rho \delta u + c^{\mathrm{u}}_\rho - q^{\mathrm{u}}_{\mathrm{subgoal}}\|$ ;
4 return $f(q_{\mathrm{nearest}}, q^{\mathrm{a}}_{\mathrm{nearest}} + \varepsilon \cdot \delta u^\star / \|\delta u^\star\|)$ ;
```

C. Contact Sampling

A node q where robots and objects are far from making contacts hinders the growth of the RRT tree for two reasons. First, such nodes are considered far away from most sampled subgoals under the local Mahalanobis distance metric (Fig. 8d). As a result, adding such nodes to the tree hardly improves coverage of the state space. Moreover, in the event that such a node is chosen by `Nearest`, the `Extend` operation that follows often results in other unfavorable configurations.

To reduce the number of such nodes in $\mathcal{T}$ and encourage exploration during tree growth, the `Extend` operation is replaced, with some probability, by a new operation called `ContactSample`. `ContactSample` takes q_{nearest} as input, and creates another node with a better local metric by fixing $q^{\mathrm{u}}_{\mathrm{nearest}}$ and finding an informative $q^{\mathrm{a}}_{\mathrm{nearest}}$ that makes contact.

The `ContactSample` operation is essential for adequate exploration of the robot’s state space, and needs to be designed differently for different robots. Here we summarize our implementation of the contact samplers for the systems in Fig. 10, which are used in our experiments in later sections. In PlanarPusher and PlanarHand, we can sample contact points on the object surface and solve for robot joint angles using inverse kinematics. For systems with the Allegro hand, we pick a random direction in the hand configuration space using EigenGrasps [65], and simulate closing the hand along that direction until contact is established.

Contact sampling introduces non-physical behavior where the robot teleports from one configuration to another. This is not a problem when the object can sustain static equilibrium without the actuated DOFs that need to teleport. For instance, in AllegroHand, when the ball is supported by the palm, the fingers are free to move around to regrasp the ball. In contrast, if AllegroHand were facing downwards, the ball would fall under gravity if it were not secured by some of the fingers. Although this is a limitation of our current implementation of contact sampling, we believe this can be resolved by a more sophisticated contact sampler which moves some of the actuated DOFs while keeping the object in static equilibrium with the rest.

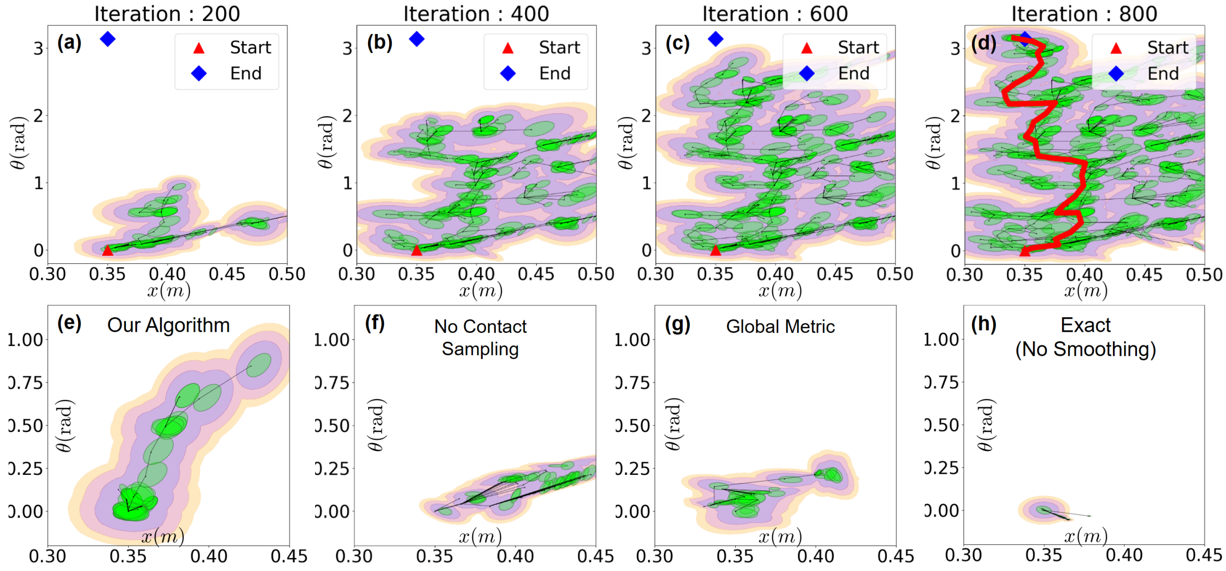


Fig. 9: (a-d) RRT trees, shown in the space of q^{u}, at different iterations of a complete run of the enhanced RRT for the PlanarHandFixedY system. The contours are the sub-level sets of the local Mahalanobis metric of the nodes. The path from start (q_{init}) to goal (q_{goal}) is highlighted in red in the final tree of **(d)**. **(e-h)** Visualization of RRT trees with the same number of nodes (50) but grown with different methods. (**e**) Tree grown with our algorithm; (**f**) without contact sampling; (**g**) using a globally uniform weighted Euclidean metric; (**h**) using exact gradients without smoothing. Note that our method achieves the best coverage of the space of q^{u}.

D. Effectiveness of Proposed Enhancements

We introduce a new system with a 2-dimensional object configuration space to illustrate the effectiveness of the proposed RRT enhancements:

- **Planar Hand with fixed y, (2,4,13)**. A simplified version of the PlanarHand system in Sec. V-B. We fix the y-coordinate of the object, so that $q^{\mathrm{u}} = (x, \theta) \in \mathbb{R}^2$ can be easily plotted on paper.

As shown in Fig. 9e, the vanilla RRT enhanced with the proposed Nearest, Extend and ContactSample achieves good coverage of the space of q^{u}, which is crucial for RRT to adequately explore the configuration space and find a path to q_{goal}. In contrast, tree growth is significantly hindered without contact sampling (Fig. 9f) or the local metric (Fig. 9g).

We also illustrate how the tree grows throughout a complete run of the enhanced RRT in Fig. 9a-d. Even with the proposed enhancements, tree growth can get stuck at times. This is characterized by a specific type of subgraph of the tree which we call a “broom”. A broom consists of one parent node with many child nodes, and is formed by repeated unsuccessful attempts to grow towards different subgoals from the same parent node. The occasional appearance of brooms is a sign that the proposed enhancements are not perfect. Nevertheless, the enhanced RRT is able to quickly branch out into empty part of the configuration space, and sufficiently cover the space as the tree grows.

E. Final Path Refinement

The final path returned by the RRT algorithm is visually plausible, yet suffers from two minor drawbacks: (**i**) RRT tends to produce randomized paths that can be shortened, and (**ii**) the big step size used in the Extend operation creates some non-physical artifacts due to Anitescu’s convex relaxation of the Coulomb friction model (Sec.III-B).

To mitigate these issues, we refine the RRT plan using trajectory optimization [37], [66] and short-cutting [67]. We first divide the RRT path into segments punctuated by ContactSample operations. We call these segments contact-rich as they involve contact-based interactions between the object and the robot. We shortcut the sequence of trajectories by (**i**) removing consecutive ContactSample steps, and (**ii**) truncating each segment if there is no movement in q^{u}. Then, for each contact-rich segment, we run trajectory optimization (Alg.1) with a smaller time step h, using the RRT path segment as the initial guess. This not only smooths the final path, but also ensures that each trajectory segment is more physically realistic. Finally, we connect adjacent contact-rich segments with a collision-free robot trajectory created by a collision-free RRT. We assume that the object configuration remains unchanged during the collision-free segment.

We find that combining these two strategies is effective in creating shorter and more physically realistic trajectories.

VIII. Planning Results & Discussion

In this section, we apply our algorithm on difficult 3D contact-rich manipulation problems previously only tackled by heavy offline approaches in RL [2], [38], and illustrate that we can generate plans on the order of a minute of online compute time, all on the CPU, which shows the efficacy of our method. The experiments in this section are designed to validate the following four hypotheses.

1) Using smooth surrogate greatly improves the performance over using the exact dynamics.
2) The equivalence of smoothing schemes establishes that analytic and randomized smoothing will have similar levels of performance empirically, with analytic smoothing showing superior computation time.
3) Using the Mahalanobis distance metric improves performance over a globally uniform distance metric.

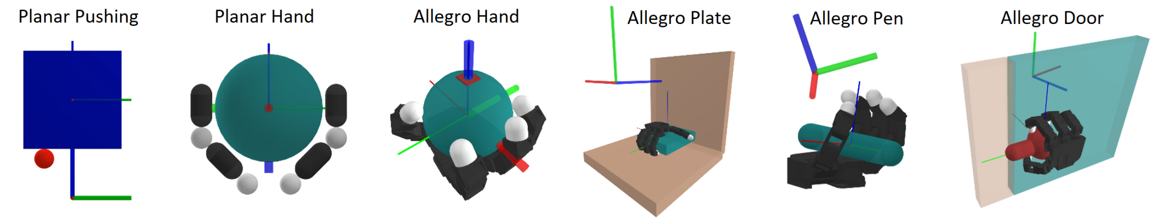


Fig. 10: Tasks for RRT. Similar to Fig. 7b, the thicker frame denotes the goal, and the thinner frame the initial configuration of the object.

4) Contact sampling greatly aids sample efficiency.

A. Experiment Setup

To test the efficacy of our algorithm and our hypotheses, we run our algorithm to reach more challenging goals than the trajectory optimization examples in Sec. V-B1, as well as on 3 more contact-rich tasks on 3 new systems defined below.

1) **Pen Placement (6,19,24)**. The robot hand move the pen [38] to the desired configuration.
2) **Plate Pickup (6,19,42)**. The robot has to exploit the external contact between the plate and the wall [10], showing extrinsic dexterity [39].
3) **Door Opening (2,19,22)** [38] involves reasoning about a constrained system, where the handle must be rotated first before the door can be pushed open.

The definition of the number tuples is identical to Sec.V-B1.

The contact-rich planning tasks are illustrated in Fig. 10. We design the tasks so that solving any of them with a single run of trajectory optimization is expected to fail due to the non-convexity of the problem.

To compare our algorithm with different baselines, we rate the quality of planners using two metrics:

1) **Iteration vs. Minimum distance to goal**. we measure the distance between the goal and the tree, defined by $\min_{q\in\mathcal{V}}\|q^{\mathrm{u}} - q^{\mathrm{u}}_{\mathrm{goal}}\|$ for every iteration. A successful planning algorithm would eventually reach the vicinity of the goal asymptotically, driving this metric to zero.
2) **Iteration vs. Packing Ratio**. To characterize the *exploration* performance of RRT, we do a Monte-Carlo estimation of the *packing ratio*, which is defined as the volume of the space occupied by the reachability ellipses, divided by the total volume of some workspace limit for the unactuated objects. The workspace limit is the set from which subgoals are sampled when running RRT. More formally, we define the numerator as

$$V_{\text{reachable}} = \textbf{vol}\big(\{q^{\mathrm{u}} | \min_{\bar{q}\in\mathcal{V}} d^{\mathrm{u}}_{\rho,\gamma}(q;\bar{q}) \leq \eta\}\big) \tag{51}$$

where η is some threshold on the distance metric. The Monte-Carlo estimate of $V_{\text{reachable}}/V_{\text{workspace}}$ can be computed by drawing N samples within the workspace and counting how many of them belong to $V_{\text{reachable}}$. A good planner should asymptotically reach a ratio of 1 if the system can reach all points in the object's workspace from which $q^{\mathrm{u}}_{\text{subgoal}}$ are sampled.

For both metrics, the performance of our RRT algorithms averaged over 5 runs is plotted in Fig. 11 for different smoothing schemes, as well as ablations of different choices we made for the algorithm. In particular, we run three variations of our algorithm while ablating a crucial ingredient.

1) **Exact**. We replace the linearization of the smooth surrogate $\mathbf{B}_\rho$ with the exact linearization $\mathbf{B}$ of CQDC dynamics, used for both extension and metric computation.
2) **NoContact**. We do not allow contact sampling (Sec.VII-C) in this variant of the algorithm.
3) **Global**. Instead of the local Mahalanobis metric, we use a globally-uniform metric during the `Nearest` step of the algorithm. For our experiments, we use a carefully-chosen weighted Euclidean norm.

B. Results & Discussion

We plot the results of our experiments in Fig. 11, and display the running time of our algorithm using the three different smoothing schemes in Table III. We discuss some of our findings from the experiment, in the context of our hypotheses in the beginning of this section.

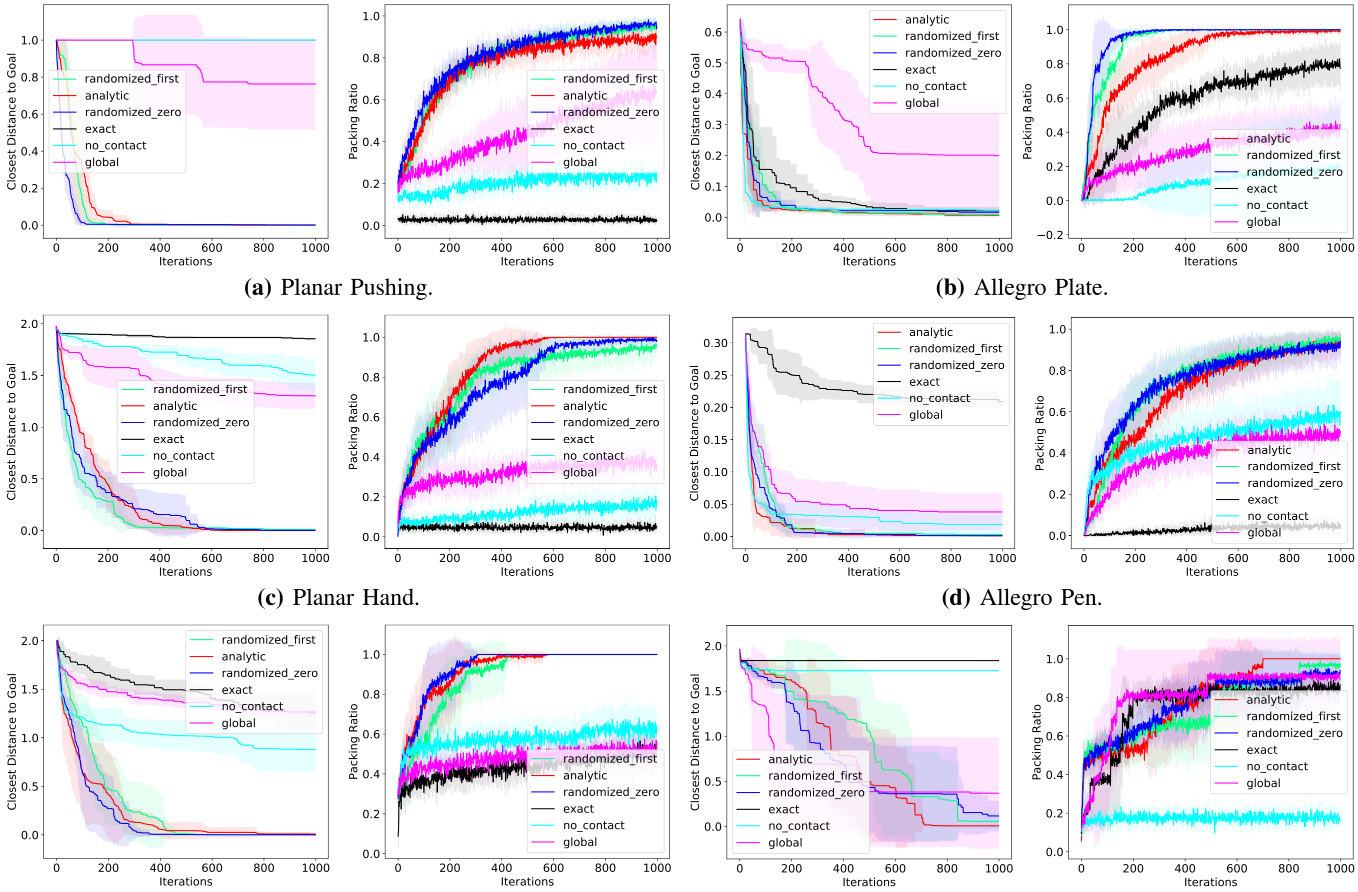
Method	PPushing	PHand	AHand	APlate	APen	ADoor
A.	3.25	10.32	31.01	117.16	24.84	12.42
RF.	7.50	21.34	80.12	161.64	86.12	34.55
RZ.	7.40	21.00	82.00	168.11	81.23	33.80

TABLE III: Running time achieved by different methods in seconds. Every trial was run for 1000 iterations. We choose to not display running time for the ablation options since they are slight variations of Analytic with comparable running times.

1) Smoothing vs. Exact: Throughout all experiments, we saw that using the exact linearization to compute the distance metric and extension results in much worse performance compared to any of the smoothing schemes, which supports our hypothesis that mode smoothing is necessary in order to solve many of the tasks.

2) Analytic vs. Randomized Smoothing: For most of the tasks, we saw no meaningful difference between analytic and randomized smoothing schemes in terms of both how fast the goal is reached and the packing ratio. This empirically supports our theory that the two smoothing schemes are equivalent methods to compute local models of surrogate dynamics. The running time in Table III, however, shows that analytic smoothing results in faster computation time as it does not require taking multiple samples.

3) Global vs. Mahalanobis Metric: Despite reasonable efforts to choose good weights for the weighted Euclidean norm, we consistently observed that the globally uniform weighted Euclidean metric resulted in much worse performance compared to the local Mahalanobis metric. This supports our hypothesis, and the findings of [6] that kino-dynamic RRT in general greatly benefits from guiding tree growth with reachability information.



(a) Planar Pushing.

(b) Allegro Plate.

(c) Planar Hand.

(d) Allegro Pen.

(e) Allegro Hand.

(f) Allegro Door.

Fig. 11: Planning performance for the tasks in Fig. 10. Results include running RRT with the enhancements proposed in Sec. VII using the three smoothing schemes from Sec. II-C, as well as the three ablation studies proposed in Sec. VIII-A.

4) Effect of Contact Sampling: For some of the tasks (e.g. AllegroPlate, AllegroPen), contact sampling was not necessary. However, for examples that require resetting the actuator into a completely different configuration to make progress (e.g. PlanarPushing, AllegroHand, PlanarHand), contact sampling greatly improves the planner's performance. Although our contact samplers are simple and can benefit from more sophisticated implementations, the experiments show that contact sampling is essential for RRT to efficiently explore through contact dynamics constraints.

IX. Sim2Real Transfer & Hardware Results

Although our planner successfully plans through our CQDC dynamics model, we further investigate if the plans can successfully transfer to real experiments. For this purpose, we run the obtained plans from Sec.VIII in *open-loop* on a higher fidelity simulator Drake [32], as well as an actual hardware setting. These experiments further shed light on the efficacy and the limitations of our proposed method.

A. Experiment Setup

1) Open-Loop Plan Transfer: Our plan consists of state and action sequences, i.e. lists of "knot" points, that are consistent with the CQDC dynamics. We first divide this plan into individual segments ($\{q_{k,\mathrm{sim}}\}_{k=0}^{K}, \{u_k\}_{k=0}^{K-1}$), punctuated by the `ContactSample` operation.

Next, we convert the knot points ($\{q_{k,\mathrm{sim}}\}_{k=0}^{K}, \{u_k\}_{k=0}^{K-1}$) into state and action trajectories $q_{\mathrm{sim}} : [0,T] \rightarrow \mathbb{R}^{n_\mathrm{u}+n_\mathrm{a}}$ and $u : [0,T] \rightarrow \mathbb{R}^{n_\mathrm{a}}$ using first-order hold[2]. Here T denotes the duration of the trajectories in seconds. Rolling out $u(\cdot)$ on the real dynamics gives $q_{\mathrm{real}} : [0,T] \rightarrow \mathbb{R}^{n_\mathrm{u}+n_\mathrm{a}}$, which is compared against $q_{\mathrm{sim}}(\cdot)$ to evaluate the sim2real performance.

2) Evaluation Metrics: To evaluate the performance of sim2real transfer, we first define the mean error $\Delta(\cdot,\cdot)$ between the two trajectories $q^{\mathrm{u}}_{\mathrm{sim}}(\cdot)$ and $q^{\mathrm{u}}_{\mathrm{real}}(\cdot)$ as

$$\Delta(q^{\mathrm{u}}_{\mathrm{sim}}, q^{\mathrm{u}}_{\mathrm{real}}) := \frac{1}{T}\int_0^T d\left(q^{\mathrm{u}}_{\mathrm{sim}}(t), q^{\mathrm{u}}_{\mathrm{real}}(t)\right) \mathrm{d}t \tag{52}$$

where $d(\cdot,\cdot)$ is the Euclidean 2-norm for position (in meters), and the absolute change in angle for orientation (in radians). For 3D, this change of angle comes from axis-angle rotations.

In addition, we expect that the metric Δ will depend on how much movement is inside the reference trajectory of the plan. To account for this scaling, we normalize Δ by dividing it by

[2] we connect adjacent knot points with linear interpolation for positions and joint angles, or spherical linear interpolation (Slerp) for 3D orientations.

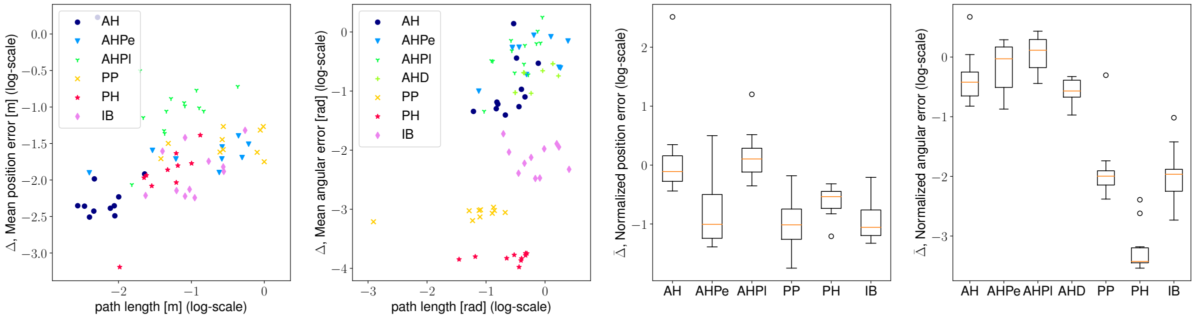


Fig. 12: Plots for sim2real performance of our CQDC dynamics, evaluated on the plans of Sec.VIII. **First Two Columns**: Scatter plot of mean error Δ vs. path length for position (first column) and orientation (second column). Each dot in the plot represents one segment trajectory. **Last Two Columns**: Box plot for the normalized error $\bar{\Delta}$ for positions (third column) and orientation (fourth column). Note that the mean in orange corresponds to the slope of the graph in the first two columns. Finally, we note that the AllegroHandDoor (AHD) example only consists of orientation, and has no position plot. Readers are highly encouraged to watch the accompanying video for the qualitative behavior of the actual segment trajectories in this plot. System abbreviations: AH(AllegroHand), AHPe(AllegroPen), AHPl (AllegroPlate), PP(PlanarPushing), PH(PlanarHand), AHD (AllegroDoor), IB(IiwaBimanual)

the length of the trajectory in the original plan, and denote the normalized error as $\bar{\Delta}$:

$$\bar{\Delta}(q^{\mathrm{u}}_{\mathrm{sim}}, q^{\mathrm{u}}_{\mathrm{real}}) := \frac{\Delta(q^{\mathrm{u}}_{\mathrm{sim}}, q^{\mathrm{u}}_{\mathrm{real}})}{L(q^{\mathrm{u}}_{\mathrm{sim}})}, \tag{53}$$

where the denominator computes the path length of $q^{\mathrm{u}}_{\mathrm{sim}}$:

$$L(q^{\mathrm{u}}_{\mathrm{sim}}) := \int_0^T \|\dot{q}^{\mathrm{u}}_{\mathrm{sim}}(t)\|_2 \,\mathrm{d}t = \sum_{k=0}^{K-1} d(q^{\mathrm{u}}_{k+1,\mathrm{sim}}, q^{\mathrm{u}}_{k,\mathrm{sim}}). \tag{54}$$

This normalization also takes into account the inherent scales of the system, and makes $\bar{\Delta}(\cdot,\cdot)$ a dimensionless quantity. For each system in Fig. 10, we obtain at least 10 segments and evaluate our error metrics.

3) Simulation Setup: We transfer the examples of Fig.10 into Drake [32], which utilizes a full second-order dynamics model with a sophisticated and accurate contact solver [68]. The collision geometries, robot controller stiffness and coefficients of friction are kept consistent between the CQDC dynamics and Drake.

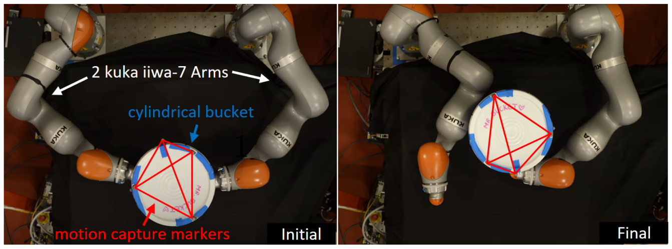


Fig. 13: Hardware for the IiwaBimanual setup, where the goal is to rotate the bucket by 180°. The left and right pictures correspond to the initial state and the final state after the open-loop plan execution. The lines between motion capture markers are connected to illustrate the change of pose in the bucket. Readers are encouraged to watch the accompanying video for the full execution.

4) Hardware Setup: To verify results on actual hardware, we create a variant of the PlanarHand environment, where the object is replaced by a bucket, and 2 Kuka iiwa arms are used for the actuators. We name this environment IiwaBimanual. We utilize a motion capture system to estimate the state of the bucket in order to compare the two trajectories of $q^{\mathrm{u}}_{\mathrm{sim}}(\cdot)$ and $q^{\mathrm{u}}_{\mathrm{real}}(\cdot)$. Our setup is illustrated in Fig. 13.

B. Results & Discussion

We plot the results of our experiments in Fig.12. While 2D systems such as PlanarPushing, PlanarHand, and IiwaBimanual display low error and good sim2real transfer, 3D systems such as AllegroHand, AllegroPlate, AllegroPen and AllegroDoor show larger error. To better understand the discrepancy of sim2real performance on different systems, we visualized trajectories from all systems by overlaying $q^{\mathrm{u}}_{\mathrm{real}}$ on top of q_{sim} (see accompanying video).

From the visualizations, it is clear that on *all* systems, there exists a persistent *phase* difference between $q^{\mathrm{u}}_{\mathrm{real}}$ and $q^{\mathrm{u}}_{\mathrm{sim}}$: $q^{\mathrm{u}}_{\mathrm{real}}$ tends to lag $q^{\mathrm{u}}_{\mathrm{sim}}$ when the robot accelerates, and lead when the robot decelerates. This is not surprising, as the CQDC dynamics that generates $q^{\mathrm{u}}_{\mathrm{sim}}$ is inherently a first-order system, whereas $q^{\mathrm{u}}_{\mathrm{real}}$ is generated from second-order dynamics. On trajectory segments with good sim2real performance, this only results in harmless oscillations of $q^{\mathrm{u}}_{\mathrm{real}}$ around $q^{\mathrm{u}}_{\mathrm{sim}}$. In these cases, we believe that the good sim2real performance validates our contact model. We now discuss other cases where sim2real is not as effective.

1) Violation of Quasi-dynamic Assumption: The quasi-dynamic assumption implies that objects are quickly stopped by damping when they are not "pushed around" by the robot. This is true on 2D systems, as the friction patch between the object and its supporting surface is persistent and always provides enough damping to bring the object to still.

However, the necessary damping to uphold the quasi-dynamic assumption does not always exist on 3D systems. For instance, in AllegroHand and AllegroHandPen, the point contact between the object and the palm provides very little damping. Most of the damping comes from the finger joints when the fingers are opposing the object's motion. Therefore, when the grasp on the object is not tight, the object can roll quite far from the planned trajectory or even off the palm.

2) Missed Contacts: Due to the non-smooth nature of contact dynamics, small discrepancies in object trajectory caused by the phase gap can lead to the robot completely missing contacts with the object. For example, we observed that in AllegroHandPlate or AllegroHandDoor, some grasps that were valid under the CQDC dynamics no longer succeeded in holding the object in place in Drake. The consequence of these failed grasps is that plates are dropped on the table in AllegroHandPlate, and door handles are missed in AllegroHandDoor.

3) Necessity of Stabilization & Robustification: These results tell us that plans suggested by the CQDC dynamics can give high-level directions, but its open-loop execution may not succeed under second-order dynamics with high velocities and low amount of damping. We believe that tracking this high-level plan requires low-level feedback controllers that can stabilize to the plan, and actively enforce the closed-loop system to be quasi-dynamic. We also believe that the high-level planer can benefit from robustness objectives such as encouraging grasps that are considered good under classical grasping metrics. Such grasps will increase the control authority of the robot over the object, thereby providing sufficient damping and decreasing the chances of dropping the object. We leave these as promising directions for future work.

X. Conclusion

We motivated our work by noting the stark contrast between the empirical effectiveness of RL for contact-rich settings and the struggle of the model-based methods. By identifying the pitfalls in the existing model-based methods for planning, understanding how RL was able to alleviate such pitfalls, and resolving them with model-based techniques, we have shown that traditional model-based approaches can be effective in tackling planning for contact-rich manipulation. Compared to existing tools in RL which use heavy offline computation in the order of hours or days, our contribution offers a powerful alternative in the spectrum of solutions by enabling efficient online planning in the order of a minute while being generalizable with respect to environments and tasks. We recap some of our contributions that enabled this process.

We first pointed out the pitfall of explicit enumeration and consideration of modes in model-based methods, whereas RL has smoothed them out stochastically. Through Sec. II, we formalized the process of smoothing for functions. This allowed us to view randomized smoothing in RL as a computational alternative to analytic smoothing used in existing model-based literature that both approximate local models of the smooth surrogate. This connection was made more concrete in the setting of contact dynamics in Sec. IV-C.

Next, we have pointed out another weakness in model-based approaches where transients of second-order dynamics lead to myopic linearizations which are uninformative about long-term planning. As a solution to this weakness, we have presented CQDC, a novel formulation of contact model that is convex, differentiable, and quasi-dynamic. Through a number of theoretical arguments and empirical studies, we have shown the efficacy of our contact model. We have further shown that by inspecting the structure in our proposed model, we can analytically smooth out the contact dynamics with a log-barrier relaxation. With experiments, we have shown that our method of analytic smoothing has computational benefits over randomized smoothing.

Finally, we observed that existing model-based methods that rely on smoothing have been tied to local trajectory optimization. Due to their weakness to local minima, they have been less effective in difficult problems compared to RL-based approaches that attempt to perform global search. On the other hand, the SBMP methods for contact-rich systems have explicitly considered contact modes which fall into the pitfall of mode enumeration. Our contribution fills in a gap in existing methods by combining mode smoothing with RRT, where local approximation to the smooth surrogate was used to guide the exploration process of RRT via the local Mahalanobis metric.

By combining these three contributions, we have enabled efficient global motion planning for highly contact-rich and high-dimensional systems that were previously not achievable by existing model-based or RL-based methods. We believe that in the future, a highly optimized version of our planner can be used to perform real-time motion planning, or be used to guide policy search. With this capability, we hope to enable robots to find contact-rich plans online in previously unseen environments within seconds of planning time.

References

[1] Andrychowicz *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

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

[3] A. Nagabandi, K. Konoglie, S. Levine, and V. Kumar, "Deep dynamics models for learning dexterous manipulation," 2019.

[4] F. Clarke, *Optimization and Nonsmooth Analysis*. Wiley New York, 1983.

[5] J. Burke, F. E. Curtis, A. Lewis, M. Overton, and L. Simões, *Gradient Sampling Methods for Nonsmooth Optimization*, 02 2020, pp. 201–225.

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

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

[8] S. Haddad and A. Halder, "Anytime ellipsoidal over-approximation of forward reach sets of uncertain linear systems," 2021.

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

[10] X. Cheng, E. Huang, Y. Hou, and M. T. Mason, "Contact mode guided motion planning for quasidynamic dexterous manipulation in 3d," *arXiv preprint arXiv:2105.14431*, 2021.

[11] B. Aceituno-Cabezas and A. Rodriguez, "A global quasi-dynamic model for contact-trajectory optimization," in *Robotics: Science and Systems (RSS)*, 2020.

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

[13] F. R. Hogan and A. Rodriguez, "Feedback control of the pusher-slider system: A story of hybrid and underactuated contact dynamics," in *Algorithmic Foundations of Robotics XII*. Springer, 2020, pp. 800–815.

[14] E. Huang, X. Cheng, and M. T. Mason, "Efficient contact mode enumeration in 3d," in *International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2020, pp. 485–501.

[15] 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.

[16] H. J. T. Suh, T. Pang, and R. Tedrake, "Bundled gradients through contact via randomized smoothing," *IEEE Robotics and Automation Letters*, pp. 1–1, 2022.

[17] H. J. T. Suh, M. Simchowitz, K. Zhang, and R. Tedrake, "Do differentiable simulators give better policy gradients?" 2022.

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

[19] J. Duchi, P. Bartlett, and M. Wainwright, "Randomized smoothing for stochastic optimization," *SIAM Journal on Optimization*, vol. 22, 03 2011.

[20] I. Mordatch, Z. Popović, and E. Todorov, "Contact-invariant optimization for hand manipulation," in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, 2012, pp. 137–144.

[21] Z. Manchester and S. Kuindersma, "Variational contact-implicit trajectory optimization," in *Robotics Research*. Springer, 2020, pp. 985–1000.

[22] Z. Huang, Y. Hu, T. Du, S. Zhou, H. Su, J. B. Tenenbaum, and C. Gan, "Plasticinelab: A soft-body manipulation benchmark with differentiable physics," *arXiv preprint arXiv:2104.03311*, 2021.

[23] T. A. Howell, S. L. Cleac'h, J. Z. Kolter, M. Schwager, and Z. Manchester, "Dojo: A differentiable simulator for robotics," *arXiv preprint arXiv:2203.00806*, 2022.

[24] M. Anitescu, "Optimization-based simulation of nonsmooth rigid multibody dynamics," *Mathematical Programming*, vol. 105, no. 1, pp. 113–143, 2006.

[25] 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)*. IEEE, 2021, pp. 6614–6620.

[26] A. Castro, F. Permenter, and X. Han, "An unconstrained convex formulation of compliant contact," *arXiv preprint arXiv:2110.10107*, 2021.

[27] 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.

[28] D. E. Stewart, "Rigid-body dynamics with friction and impact," *SIAM review*, vol. 42, no. 1, pp. 3–39, 2000.

[29] A. Agrawal, S. Barratt, S. Boyd, E. Busseti, and W. M. Moursi, "Differentiating through a cone program," *arXiv preprint arXiv:1904.09043*, 2019.

[30] M. T. Mason, *Mechanics of robotic manipulation*. MIT press, 2001.

[31] N. Chavan-Dafle, R. Holladay, and A. Rodriguez, "In-hand manipulation via motion cones," 2019.

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

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

[34] A. Ö. Önol, R. Corcodel, P. Long, and T. Padır, "Tuning-free contact-implicit trajectory optimization," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1183–1189.

[35] S. M. LaValle *et al.*, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[36] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 7681–7687.

[37] H. J. T. Suh, X. Xiong, A. Singletary, A. D. Ames, and J. W. Burdick, "Energy-efficient motion planning for multi-modal hybrid locomotion," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 7027–7033.

[38] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," 2018.

[39] N. C. Dafle *et al.*, "Extrinsic dexterity: In-hand manipulation with external forces," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1578–1585.

[40] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015.

[41] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 3, 05 1992.

[43] Z. Yang, K. Balasubramanian, Z. Wang, and H. Liu, "Estimating high-dimensional non-gaussian multiple index models via stein's lemma," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V.

[42] C. Stein, P. Diaconis, S. Holmes, and G. Reinert, "Use of exchangeable pairs in the analysis of simulations," vol. 46, 01 2004.

Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[44] S. P. Bangaru, J. Michel, K. Mu, G. Bernstein, T.-M. Li, and J. Ragan-Kelley, "Systematically differentiating parametric discontinuities," *ACM Trans. Graph.*, vol. 40, no. 4, Jul. 2021.

[45] T. Pang and R. Tedrake, "Easing reliance on collision-free planning with contact-aware control," *arXiv preprint arXiv:2109.09846*, 2021.

[46] B. Landry, Z. Manchester, and M. Pavone, "A differentiable augmented lagrangian method for bilevel nonlinear optimization," *arXiv preprint arXiv:1902.03319*, 2019.

[47] V. Kurtz and H. Lin, "Contact-implicit trajectory optimization with hydroelastic contact and ilqr," *arXiv preprint arXiv:2202.13986*, 2022.

[48] 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*. IEEE, 2012, pp. 5026–5033.

[49] R. Kolbert, N. Chavan-Dafle, and A. Rodriguez, "Experimental validation of contact dynamics for in-hand manipulation," in *International Symposium on Experimental Robotics*. Springer, 2016, pp. 633–645.

[50] N. Hogan, "Impedance control: An approach to manipulation: Part i—theory," 1985.

[51] M. Anitescu, J. F. Cremer, and F. A. Potra, "Formulating three-dimensional contact dynamics problems," *Journal of Structural Mechanics*, vol. 24, no. 4, pp. 405–437, 1996.

[52] M. ApS, *Mosek Modeling Cookbook*, 2021. [Online]. Available: https://docs.mosek.com/MOSEKModelingCookbook-letter.pdf

[53] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, "SCS: Splitting conic solver, version 3.2.1," https://github.com/cvxgrp/scs, Nov. 2021.

[54] J. Li, G. Daviet, R. Narain, F. Bertails-Descoubes, M. Overby, G. E. Brown, and L. Boissieux, "An implicit frictional contact solver for adaptive cloth simulation," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–15, 2018.

[55] 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.

[56] J. Carpentier and N. Mansard, "Analytical derivatives of rigid body dynamics algorithms," in *Robotics: Science and systems (RSS 2018)*, 2018.

[57] G. Guennebaud, B. Jacob *et al.*, "Eigen v3," http://eigen.tuxfamily.org, 2010.

[58] S. Andrews, K. Erleben, and Z. Ferguson, "Contact and friction simulation for computer graphics," in *ACM SIGGRAPH 2022 Courses*, ser. SIGGRAPH '22. New York, NY, USA: Association for Computing Machinery, 2022.

[59] S. L. Cleac'h, M. Schwager, Z. Manchester, V. Sindhwani, P. Florence, and S. Singh, "Single-level differentiable contact simulation," 2022.

[60] L. Montaut, Q. L. Lidec, A. Bambade, V. Petrik, J. Sivic, and J. Carpentier, "Differentiable collision detection: a randomized smoothing approach," 2022.

[61] P. Varin and S. Kuindersma, *A Constrained Kalman Filter for Rigid Body Systems with Frictional Contact*, 05 2020, pp. 474–490.

[62] T. Howell, S. Le Cleac'h, S. Singh, P. Florence, Z. Manchester, and V. Sindhwani, "Trajectory optimization with optimization-based dynamics," *IEEE Robotics and Automation Letters*, 2022.

[63] 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.

[64] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[65] M. T. Ciocarlie, C. Goldfeder, and P. K. Allen, "Dexterous grasping via eigengrasps : A low-dimensional approach to a high-complexity problem," 2007.

[66] M. Toussaint, "Logic-geometric programming: An optimization-based approach to combined task and motion planning," in *IJCAI*, 2015, pp. 1930–1936.

[67] R. Geraerts and M. H. Overmars, "Creating high-quality paths for motion planning," *The International Journal of Robotics Research*, vol. 26, no. 8, pp. 845–863, 2007.

[68] A. M. Castro, A. Qu, N. Kuppuswamy, A. Alspach, and M. Sherman, "A transition-aware method for the simulation of compliant contact with regularized friction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1859–1866, apr 2020.