SDRS: Shape-Differentiable Robot Simulator

Xiaohan Ye[1], Xifeng Gao[2], Kui Wu[2], Zherong Pan[2†], and Taku Komura[1†]

***Abstract*—Robot simulators are indispensable tools across many fields, and recent research has significantly improved their functionality by incorporating additional gradient information. However, existing differentiable robot simulators suffer from non-differentiable singularities, when robots undergo substantial shape changes. To address this, we present the Shape-Differentiable Robot Simulator (SDRS), designed to be differentiable under significant robot shape changes. The core innovation of SDRS lies in its representation of robot shapes using a set of convex polyhedrons. This approach allows us to generalize smooth, penalty-based contact mechanics for interactions between any pair of convex polyhedrons. Using the separating hyperplane theorem, SDRS introduces a separating plane for each pair of contacting convex polyhedrons. This separating plane functions as a zero-mass auxiliary entity, with its state determined by the principle of least action. This setup ensures global differentiability, even as robot shapes undergo significant geometric and topological changes. To demonstrate the practical value of SDRS, we provide examples of robot co-design scenarios, where both robot shapes and control movements are optimized simultaneously.**

***Index Terms*—Differentiable Simulation, Articulated Body, Robot Design, Policy Search.**

I. Introduction

Robot design and control stand as two paramount areas of research. In robot design, engineers manipulate the hardware specifications of a robot to equip it with the desired motion capabilities, while in robot control, algorithms compute the requisite motions and control signals for accomplishing a wide array of tasks. Although both domains share the overarching objective of task fulfillment, they have traditionally been treated as separate stages in the blueprint of a robotic system. Robot designers often rely heavily on their experience and observations to fine-tune design parameters [1, 2, 3], while robot controllers are typically crafted with the assumption of fixed robot designs [4, 5]. Unfortunately, recent studies [6, 7, 8, 9] have revealed that neglecting the inherent interdependence between these two stages can inevitably result in sub-optimal designs.

The divide between design and control has increasingly captured the attention of researchers. Various techniques have emerged to bridge this gap, and among these, a promising approach is robot co-design, where both the robot's design and controller parameters are jointly optimized. A significant challenge in such applications lies in the expansive search space, encompassing discrete decision variables for the robot's topology and continuous ones for its geometry and controller parameters. To derive meaningful robot designs, existing methods often resort to aggressive constraints on the robot's design space and employ exhaustive sampling-based optimization strategies, such as Bayesian optimization [8, 10] and Monte-Carlo tree search [7]. The necessity of constraining the design space arises because these sampling-based optimizers grapple with the curse of dimensionality, leading to exponential complexity as the design space grows. Consequently, this restriction on the design space can, once again, result in sub-optimal designs.

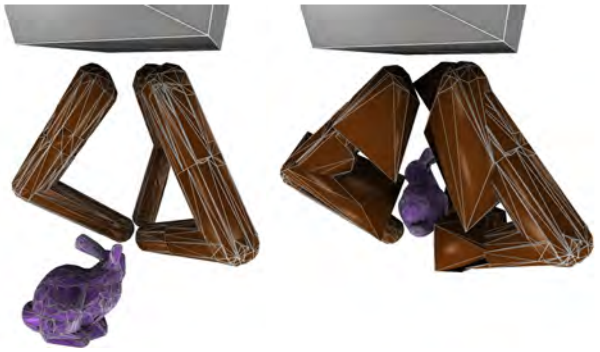

Fig. 1: Starting from an initial guess (left), we optimize the shape of gripper to firmly grasp the Stanford bunny (right).

Recent advancements in sensitivity analysis [6] and robot simulation [9] have introduced a gradient-based approach as an alternative to traditional sampling-based methods. These algorithms use differentiable simulators to propagate gradient information throughout robot simulations, allowing this information to guide a variety of upstream applications, including controller optimization [11], behavior cloning [12], and robot system identification [13]. Unlike traditional methods, gradient-based algorithms are not constrained by the curse of dimensionality, enabling them to scale across large design spaces and optimize complex aspects of robot link configurations. Despite their promising results, current differentiable simulators are either incapable of modeling significant changes to geometric shapes, restricting the changes to small localized neighborhoods, or not provably differentiable.

We present, SDRS, a guaranteed collision-free robot simulator that can differentiate through significant geometric and topologic shape changes to robot links. Our approach is founded on two cornerstones. First, we introduce a novel parameterization for the shape of robot links using the union of convex polyhedrons in the V-representation [14]. This expressive representation allows both geometric and topologic shape changes in relatively low-dimensional continuous parametric shapes. Second, we propose a penalty-based collision constraint between pairs of convex hulls based on

[†] indicates corresponding author. (Corresponding author: Zherong Pan and Taku Komura.) [2]LIGHTSPEED. {gxf.xisha@gmail.com, walker.kui.wu@gmail.com, zherong.pan.usa@gmail.com}. [1]The Department of Computer Science, Hong Kong University. {u3010417@connect.hku.hk, taku@cs.hku.hk}

the separating hyperplane theorem [15]. We show that this formulation can be extended to model contact mechanics by treating the separating plane as a zero-mass auxiliary entity. Our formulation avoids the need for explicit contact point detection between convex hulls, thereby eliminating non-differentiable singularities. We show that our formulation is globally differentiable under sufficiently small timestep sizes, ensuring consistently well-defined gradient information. To underscore the practical utility of our simulator, we showcase its applications over a row of robot co-design scenarios as shown in Figure 1.

II. Related Work

We review related works on robot co-design, differentiable simulation, and contact mechanics.

a) Robot Co-design: Joint optimization strategies for robot co-design have been developed to fine-tune both the physical and geometric parameters, as well as controller parameters during testing, all aimed at achieving optimal task performance. The automatic, joint optimization functionality complements the semi-automatic, human-assisted design pipeline [16, 17, 18] to form a complete robot design and manufacturing workflow. Much of the existing research, such as that by [7, 8, 10, 19, 20], has depended on gradient-free, sampling-based methods. A significant limitation of these approaches is their limited scalability. To reduce these costs, Bayesian optimization [8, 10] has been used to model the landscape of low-level objectives and to generate new samples that minimize fitting errors. Additionally, Zhao et al. [7] capitalized on the structural insight that open-loop articulated robots have a tree-like topology, similar to the decision trees in MCTS, enabling quick assessment and pruning of sub-optimal design candidates. Despite these enhancements, sampling-based algorithms struggle to scale beyond a few tens of parameters.

On the other hand, gradient-based methods [6, 9, 21, 22] have only recently come into prominence. Thanks to their efficient gradient evaluation algorithms, these methods incur lower iterative costs and are capable of scaling to high-dimensional decision spaces. Initial applications of these methods [6] were confined to low-dimensional design spaces, including variables such as robot link lengths and actuator positions. More recent developments [9, 21] have broadened the scope of these techniques, enabling optimization of more complex robot link shapes. However, these advanced methods still rely on mesh-based shape representations with the fixed topology, which limit changes to minor geometric adjustments.

b) Differentiable Simulation: The realm of optimal control has witnessed substantial advancements through the adoption of differentiable simulators. Their application scope has rapidly broadened, encompassing control not only over low-dimensional rigid bodies [23] and articulated bodies [24] but also extending to the control of high-dimensional deformable objects, such as elastic objects [25, 26, 27, 28, 29, 30], fluids [31], cloth [32], and general multi-physics systems [33]. Furthermore, an increasing body of research has integrated differentiable simulators into downstream end-to-end learning algorithms to support various applications, including behavior cloning [12], system identification [13, 34], and controller optimization [11]. While these applications solely require a simulator to be differentiable with respect to state and control parameters, the field of co-design necessitates the calculation of gradients with respect to link shapes.

Addressing this need, Xu et al. [9] expanded upon prior work [35] by introducing additional design gradients using a cage-based robot shape deformation model. While promising results have been presented, their deformable model is based on a mesh of fixed topology, which inherently does not allow significantly shape changes such as modifications to the link topology. We are also aware of the differentiable Material Point Method (MPM) [11], which was originally used to simulate soft robots, can be adapted to approximately simulate rigid bodies by using a large stiffness. Using particle-based representations, MPM can differentiate through large geometric and even topological changes. Unfortunately, particle-based representations are inherently incapable of representing smooth and flat surfaces unless a dense set of particles is used, making them rather inefficient for co-design applications.

Notably, despite claims of proposing differentiable simulators in previous research, none of these simulators are rigorously differentiable concerning all input parameters. For instance, some formulations, like those presented by Avila Belbute-Peres et al. [23] and Werling et al. [36], incorporate contact modeling using complementary constraints with non-smooth transitions between contact modes. Although complementary constraints can be smoothened as shown in [37, 38], these methods often require maintaining discrete contact points for constraint instantiation, which can undergo non-smooth variations due to the change of robot link shapes.

Complementing the works on differentiable simulators, physics-constrained trajectory optimization [39, 40] formulates the equations of motion as a set of (in)equality constraints, which are enforced while optimizing high-level objective functions. These methods are computationally more efficient since they avoid the need for explicit constraint satisfaction at every iteration and have been successfully applied to co-design problems, as demonstrated in [41, 42]. However, these techniques rely on additional mechanisms to ensure constraint satisfaction, which are not guaranteed to succeed.

c) Contact Mechanics: Collisions and contacts play a significant role in shaping various phenomena within robot simulations, yet they also constitute a primary source of discontinuity and non-smooth behavior. The efficacy of collision detection techniques is heavily reliant on the chosen geometric representation. Among the two most prevalent representations are volumetric distance field [43] and surface meshes [44], both of which persist as fundamental components in state-of-the-art differentiable simulators [9, 35, 45]. These techniques identify points of penetration and employ penalty forces to mitigate the depth of penetration. However, these discrete collision points can undergo non-smooth transitions as the shapes of robot links change. An alternative collision resolution approach revolves around complementary constraints [36], a method well-known for its non-smooth characteristics. Moreover, a common challenge associated with these collision techniques arises when dealing with geometrically thin objects. As they allow penetration between objects, simulators may inadvertently bypass crucial collisions by tunneling through thin objects.

Method	Geometry	Smooth	Diff.	Coll.-Free	Topo.-Change
[9, 45]	Mesh	✓	✗	✗	✗
[28]	Mesh	✓	✓	✓	✗
[11]	Particle	✗	✓	✗	✓
Ours	Convex Hull	✓	✓	✓	✓

TABLE I: We compare different articulated body simulation algorithms in terms of several features: the geometric representation, ability to represent smooth and planar surfaces, provable differentiability, provable collision-free guarantee, and differentiability under topology changes (from left to right). Our method is the first with a complete feature set.

In contrast, we are inspired by the interior point method [46, 47], wherein collisions are preemptively averted by introducing barrier functions to confine robots within collision-free spaces. However, these techniques use mesh-based representations with fixed topology for robot links, which offer a less compact representation of robot designs. Instead, we adopt a significantly more concise and expressive representation—namely, the union of convex polyhedrons in the V-representation [14]. This representation permits alterations to both the geometry and topology of robot link shapes using a relatively smaller set of decision variables. We are aware of a related recent work [48] that allows differentiable collision detection between convex hulls. However, it is still unclear how to incorporate such collision detector into the full pipeline of contact handling as required by dynamic simulations, which is the goal of our work. In summary, by comparing the features of different algorithms in Table I, we highlight that our method is the first algorithm with a complete feature set.

III. Problem Definition

In this section, we introduce the general framework of robot co-design using the novel design space proposed in Deng et al. [14] as illustrated in Figure 2. Then in Section IV, we propose our robot simulator tailored to this design space.

A. Robot Co-design

We denote the robot design as parameterized by a set of continuous variables represented as d. Furthermore, we assume the existence of a convex design space denoted as $\mathcal{D}$, where only values of $d \in \mathcal{D}$ correspond to valid robot designs. We maintain the assumption that the robot's number of degrees of freedom, i.e. the dimension of configuration space, remains invariant with respect to the robot's design. At each discrete time step t, the kinematic state of the robot is represented as the vector θ^t. Additionally, the robot is under the control of a parameterized signal u^t at time step t. With these notations in place, our simulator is defined as the following function $\theta^{t+1} = f(\theta^t, \theta^{t-1}, u^t, d)$, where the first two parameters represent the robot's kinematic state at two distinct time instances, combining to describe its dynamic state. Note that we assume θ^{t+1} is a function of form $\theta^{t+1}(u^t, u^{t-1}, \cdots, d)$, but we use the abbreviation for brevity. The overarching objective of robot co-design is to optimize both controller and design parameters through a joint optimization of a loss function $L(\theta^H)$ over a trajectory of H time steps: $\text{argmin}_{d\in\mathcal{D},c\in\mathcal{C}} L(\theta^H)$, where we assume that our control policy is represented as a differentiable function $u^t = \pi(\theta^t, \theta^{t-1}, t, c)$, which is parameterized by another set of continuous parameters c, and it operates within a convex control space denoted as $c \in \mathcal{C}$. The co-design optimization process is guided by a terminal-state differentiable cost function that estimates the quality of task completion. To solve this co-optimization, we employ the standard projected gradient method [49]. In the next section, we introduce a design space that strikes a balance between compactness and expressiveness, enabling us to make modifications to both the geometry and topology of robot links. Building upon this versatile design space, our subsequent goal is to craft a contact-aware robot simulator function, f, which demonstrates provable differentiability concerning all of its parameters. We refer to this as the SDRS simulator.

B. Design Space Parameterization

The design of robots integrates several aspects, such as joint parameters, joint limits, and motor thrusts. Previous studies such as [9] have demonstrated that articulated body simulators are differentiable with respect to these features. We primarily focus on the parametrization of robot link shapes. Drawing on recent advances from Deng et al. [14], we model the ith robot link as a combination of N smoothed convex polyhedrons, each defined within a reference frame and denoted as $H_{i1}, \cdots, H_{iN}$. Each polyhedron is characterized by a V-representation, defined by the convex combination of a set of M vertices: $H_{ij} \triangleq \text{CH}(x_{ij}^1, \cdots, x_{ij}^M)$, where CH denotes the convex hull operation. This parametric method not only retains compactness but also offers extensive expressiveness, effectively capturing a wide range of shapes with diverse geometry and topology. Further, the SDRS allows for the adjustment of the attachment point between the ith joint and its parent joint, denoted by $\lambda(i)$, with the attachment point represented as $x_i^{\lambda(i)}$. This results in the design space of dimension $|d| = 3\#(NM+1)$ with $\#$ being the number of links.

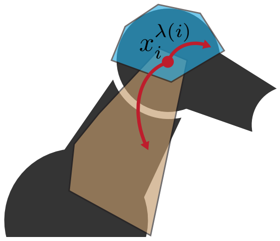

Fig. 3: The constraint to ensure that the attachment point $x_i^{\lambda(i)}$ overlaps both links.

As a potential problem, permitting the vertices to move freely may result in disconnected robot links. To address this issue, we introduce connectivity constraints. Specifically, for a pair of polyhedrons, H_{ij} and H_{ik}, connectivity is ensured by introducing an auxiliary vertex x_{ijk} and requiring that x_{ijk} is a common vertex shared by the two convex hulls. All imposed constraints are convex and consistent with our assumptions regarding $\mathcal{D}$. Similarly, we

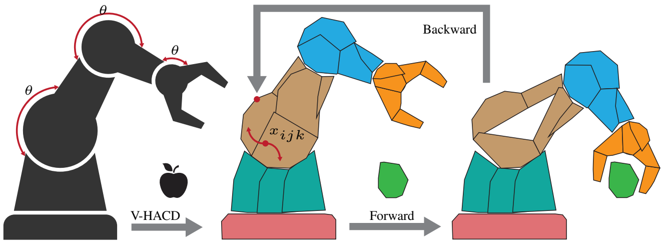

Fig. 2: We illustrate the co-design problem of a robot arm trying to reach a target apple, with the robot's configuration space being the 3 joint angles (red). We first represent each robot link shape as a set of (possibly overlapping) convex polyhedrons (middle), e.g. using V-HACD, where we use different colors for different rigid bodies. A connectivity constraint denoted as x_{ijk} is introduced for each pair of overlapping convex polyhedrons (red). Our SDRS is a differentiable robot simulator tailored to the convex-polyhedron-based representation, where we can differentiate with respect to the polyhedron vertices. Such representation allows both geometry and topology changes by continuously modifying the vertices, e.g. to drill a hole in the middle of the brown link (right).

ensure that the ith link is connected to its parent by requiring that $x_i^{\lambda(i)}$ is located within some convex hull encompassing both the ith and $\lambda(i)$th links as shown in Figure 3. Practically, users may supply mesh-based link representations, which we convert to our parametric form using approximate convex decomposition [50]. We then insert connectivity constraints between each pair of initially overlapping convex hulls, as well as between the convex hulls of the ith and $\lambda(i)$th links containing the attachment point.

IV. SDRS Formulation

In this section, we develop a collision-free robot simulator under minimal coordinates, which is tailored to the convex-polyhedron-based representation. We first delve into position-level articulated body dynamics in Section IV-A, which is formulated as an unconstrained optimization with twice-differentiable objectives. In Section IV-B, we then present a novel formulation of differentiable contact mechanics between convex polyhedrons, and we extend it to handle frictional damping in Section IV-C. Finally, we discuss the simulation algorithms and the derivative evaluation scheme in Section IV-D

A. Position-level Articulated Dynamics

For now, we omit the collisions and contacts and elucidate the dynamics of the articulated body using a position-level formulation as outlined in [51], which is a special form of physics simulation under generalized coordinates [52]. It is known that a position-level formulation can rigorously preserve collision-free properties [47, 53] via a well-defined barrier energy function, which in turn leads to our desired property of provable differentiability. We begin by formulating the forward kinematic function of the ith link. The transformation from reference to world space for this link can be defined recursively as:

$$T_i(d,\theta) \triangleq T_{\lambda(i)}(d,\theta) \begin{pmatrix} R_i^d(d) & t_i^d(d) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_i(\theta) & t_i(\theta) \\ 0 & 1 \end{pmatrix}, \quad (1)$$

where $T_{\lambda(i)}(d,\theta)$ is the transformation of the parent link, $R_i^d(d)$ and $t_i^d(d)$ define the design-dependent local transformation of attachment points of the ith link in $\lambda(i)$th reference frame, and finally $R_i(\theta)$ and $t_i(\theta)$ represent the joint transformation of the ith link. For the sake of brevity, we have omitted the timestep index and we denote x as the 4D homogeneous coordinates so that $T_i(d,\theta)x$ is the transformed coordinates in world space. Position-level dynamics hinges on the measurement of velocity in Euclidean space rather than configuration space. Specifically, we compute the acceleration of a local point x on ith link as follows:

$$\ddot{x} \triangleq \frac{1}{\Delta t^2}\left[T_i(d,\theta^{t+1}) - 2T_i(d,\theta^t) + T_i(d,\theta^{t-1})\right]x, \quad (2)$$

where Δt represents the timestep size. Note that we have employed a first-order finite-difference approximation for acceleration, which suffices for our specific application. However, higher-order approximations are also available for those seeking enhanced accuracy.

If we define $U(\bullet)$ as the world-space conservative potential energy density, we can formulate position-level dynamics as the following unconstrained minimization:

$$\begin{aligned} \theta^{t+1} &= f(\theta^t,\theta^{t-1},u^t,d) \triangleq \underset{\theta^{t+1}}{\operatorname{argmin}} \sum_{ij} I_{ij}(d,\theta^{t+1},\theta^t,\theta^{t-1},u^t) \\ I_{ij} &\triangleq \int_{x\in H_{ij}(d)} \rho_{ij}(x)\left[\frac{\Delta t^2}{2}\|\ddot{x}\|^2 + U(T_i(d,\theta^{t+1})x,u^t)\right]dx, \end{aligned} \quad (3)$$

where $\rho(x)$ represents the mass density of the robot link. As shown in [51], the above integral can be computed for a fixed robot design by pre-computing inertial-like tensors. As a remarkable feature of [51], the bilinear term describing the centrifugal and Coriolis forces can be absorbed into the above integral, inducing a concise expression of time integration and differentiation. However, when dealing with a changing integration domain, $H_{ij}(d)$, this pre-computation is no longer feasible. Instead, we make the assumption that the mass of the robot link is evenly distributed among the convex hull vertices, i.e., $\rho_{ij}(x) = \sum_{m=1}^{M}\rho\delta(x - x_{ij}^m)$, where δ represents the Dirac delta function. This allows us to simplify the integral to the following summation:

$$I_{ij} \triangleq \rho\sum_{m=1}^{M}\left[\frac{\Delta t^2}{2}\|\ddot{x}_{ij}^m\|^2 + U(T_i(d,\theta^{t+1})x_{ij}^m,u^t)\right]. \quad (4)$$

The position-level dynamics framework is highly versatile and can accommodate various external and internal force models by adding additional energy terms, $U(\bullet)$, where we further require $U(\bullet)$ to be a smooth function. In particular, we are interested in applying control signals using a stable-PD controller [54]. In this context, the control signal is defined by a desired target robot position $\theta_\star^{t+1}$ and velocity $\dot{\theta}_\star^{t+1}$, and we define it as: $u^t = \left(\theta_\star^{t+1},\dot{\theta}_\star^{t+1}\right)$. The stable-PD controller is specified by adding the following controller-energy term:

$$U_{\text{pd}}(\theta^{t+1},\theta^t,u^t) \triangleq k_p\|\theta_\star^{t+1} - \theta^{t+1}\|^2 + k_d\|\dot{\theta}_\star^{t+1} - (\theta^{t+1} - \theta^t)/\Delta t\|^2, \quad (5)$$

where k_p and k_d are the position and velocity gains, respectively.

B. Contact Mechanics for Convex Polyhedrons

Moving forward, we address the intricate aspects of (self-)collisions and contacts within our approach. Note that rigid body simulations using convex polyhedral shapes is a common practice in real-time simulators [55], but these methods need to explicitly detect and manage contact points, rendering them less amenable to differentiation. Our methodology draws inspiration from the interior point method [46, 47]. These techniques introduce a barrier energy function denoted as $U_c(\bullet)$ into the numerical optimizer, a function defined exclusively within the collision-free domain. Moreover, these barrier energies can be intentionally designed to possess sufficient smoothness, rendering them amenable to differentiation. However, a challenge arises when applying this approach to our representation of robot shapes. Indeed, contacts can occur not only between vertices but also faces and edges. With the shape changes of convex polyhedrons, these face-wise or edge-wise contacts can dynamically appear or vanish, leading to non-differentiable singular configurations.

We propose an alternative barrier energy formulation between a pair of convex polyhedrons, H_{ij} and $H_{i'j'}$, that is inspired by separating hyperplane theorem, which was previous used to handle contacts for UAV swarms in [15]. This theorem claims that, when two convex shapes exhibit no penetration, a separating plane, denoted as $p \triangleq \big(n, o\big) \in \mathbb{R}^4$, exists such that H_{ij} and $H_{i'j'}$ reside on opposite sides of p [56], which can be formally modeled through the following set of constraints:

$$\|n\| = 1 \wedge \forall m = 1, \cdots, M : \begin{cases} p^T T_i(d,\theta) x_{ij}^m \leq 0 \\ p^T T_{i'}(d,\theta) x_{i'j'}^m \geq 0 \end{cases}, \quad (6)$$

where $n \in \mathbb{R}^3$ is the normal of plane and $o \in \mathbb{R}$ is the offset. Notably, the first constraint ensures that the separating plane is well-defined with a unit-normal. Regretfully, the unit-normal constraint is non-convex and can incur significant difficulty in numerically satisfying Equation 6. To resolve this problem, Hönig et al. [15] replaced the unit-normal constraint with the convex constraint, $\|n\| \leq 1$, which is widely used in the Support Vector Machine (SVM) [57] to equivalently ensure the separation of convex sets [58].

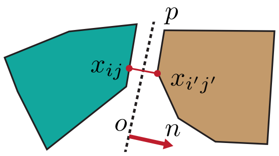

Fig. 4: The separating hyperplane (dashed) between a pair of convex polyhedrons with normal n (red). The closest distance (red) between them is realized by x_{ij} and $x_{i'j'}$.

Building upon the principles of the interior-point method, we formulate the aforementioned constraints as a smooth barrier function. Li et al. [47] proposed the clamped log-barrier function that is locally supported in $(0, s)$ where s is the user-selected support size. Instead, we use the function of $P_s(x) = (x-s)^4/x^5 \mathbf{1}(x < s)$ with $\mathbf{1}(x < s)$ being the indicator function that takes value 1 if and only if $x < s$. P_s has the similar property but is convenient for our analysis, and we derive the following contact potential for the pair of convex polyhedrons:

$$\begin{aligned} &U_{c\star}^{ij,i'j'}(d,\theta,p) \triangleq P_s(1-\|n\|)+ \\ &\sum_{m=1}^{M} \left[P_s(-p^T T_i(d,\theta) x_{ij}^m) + P_s(p^T T_{i'}(d,\theta) x_{i'j'}^m])\right]. \end{aligned} \quad (7)$$

Our contact potential effectively circumvents the need to identify face-wise or edge-wise collisions by consolidating energies directly into the vertices of the convex hulls, which are directly mapped to our parameter space. Note that the definition of $U_{c\star}^{ij,i'j'}$ remains under-determined, as it relies on the additional separating plane p. To resolve this issue, we consider the separating plane as an auxiliary physical entity characterized by zero mass as shown in Figure 4. Consequently, its presence exerts a negligible influence on other physical entities, so the entity is treated as a force and torque conductor between the two objects in contact. Since p possesses zero mass, it must experience zero force and torque, as any non-zero force or torque would result in infinite acceleration. This idea has been adopted to analyze several physical phenomena in special cases [59, 60]. However, as far as we know, this is the first time that the idea of virtual objects is adopted to derive a complete contact model. We observe that the only potential energy associated with p is the contact energy, so that p must constitute a local minimum of $U_{c\star}^{ij,i'j'}$, yielding a well-determined barrier potential: $U_c^{ij,i'j'}(d,\theta) \triangleq \min_p \; U_{c\star}^{ij,i'j'}(d,\theta,p)$ and we define the optimal separating plane as $p_\star$. In our Section VII, we show that $U_c^{ij,i'j'}$ is a well-defined, twice-differentiable contact potential that strictly prevents any intersections between the pair of polyhedron.

Our contact potential is amenable to efficient computations. As a first step, we can construct a Bounding Volume Hierarchy (BVH) for all the convex polyhedrons and $U_c^{ij,i'j'}$ only takes non-zero values when the distance between H_{ij} and $H_{i'j'}$ is less than $2s/(1-s)$ (see Section VII). By bulging each bounding box by this distance, we can accelerate energy evaluations by pruning non-overlapping pairs. When two bounding boxes overlap, we then use the GJK algorithm [61] to compute the exact distance between the two polyhedrons, along with the pair of points x_{ij} and $x_{i'j'}$ realizing the smallest distance. If the energy can be non-zero, we initialize the separating plane p to be the middle orthogonal plane between x_{ij} and $x_{i'j'}$. Next, we use the Newton's method to find the globally optimal p to compute $U_c^{ij,i'j'}$. Finally, the derivatives of our formulated contact potential can be derived using the (high-order) implicit function theorem.

C. Frictional Contact Mechanics

Expanding our contact model to incorporate frictional forces presents a unique challenge that necessitates a novel formulation. Prior simulators have traditionally employed point-wise Coulomb's frictional models, whereas our contact model operates between convex hulls and does not explicitly identify contact points. To bridge this gap, we once again turn to the separating plane, denoted as $p_\star^t = \big(n_\star^t, n_{0\star}^t\big)$, where the superscript indicates the separating plane computed at the tth timestep.

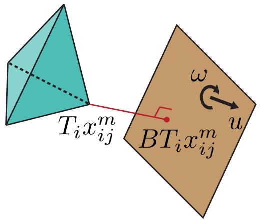

Fig. 5: We assume the separating plane is moving at an in-plane angular speed of ω and linear speed of u. Each vertex $T_i(d,\theta^t)x_{ij}^m$ is projected onto the plane as $B^{ij,i'j'}T_i(d,\theta^t)x_{ij}^m$, to compute the relative velocity.

Modeled as a physical entity with zero-mass, we make the assumption that the separating plane is undergoing tangential motion with a linear speed of $u \in R^2$ and an angular speed of $\omega \in R$. Instead of introducing frictional forces between the two convex hulls, we propose a novel approach illustrated in Figure 5: introducing frictional forces between each vertex of the convex hull and the separating plane. For instance, considering x_{ij}^m as an example, the normal force exerted on it at the tth timestep is defined as:

$$f_{ij,\perp}^m(d,\theta^t) \triangleq -\frac{\partial U_c^{ij,i'j'}}{\partial[T_i(d,\theta^t)x_{ij}^m]_3} = P_s' n_\star^t, \tag{8}$$

where we use $[\bullet]_3$ to extract the first 3 rows of homogeneous coordinates. A similar formula is applicable to the force on the other polyhedron, $f_{i'j',\perp}^m$. We introduce the notation for the tangent space basis of the separating plane: $B^{ij,i'j'}(d,\theta^t) \in \mathbb{R}^{3\times 2}$. Subsequently, we approximate the tangential velocity of x_{ij}^m using a first-order finite difference as follows:

$$\dot{x}_{ij,\parallel}^m \triangleq \frac{1}{\Delta t}[B^{ij,i'j'}]^T[T_i(d,\theta^{t+1})x_{ij}^m - T_i(d,\theta^t)x_{ij}^m]_3. \tag{9}$$

Leveraging the separating plane, we can calculate the tangential velocity of the plane at the projected point of x_{ij}^m. This is defined as:

$$\dot{p}_{ij,\parallel}^m(u,\omega) \triangleq \omega[B^{ij,i'j'}]^T[n_\star^t \times [T_i(d,\theta^t)x_{ij}^m]_3] + u. \tag{10}$$

The first and second terms above arise from tangential rotation and translation, respectively. Armed with this information, we can employ Coulomb's frictional law at the point-wise level and derive an equivalent expression for the frictional force acting on x_{ij}^m as follows:

$$f_{ij,\parallel}^m \triangleq -\mu\|f_{ij,\perp}^m(d,\theta^t)\|B^{ij,i'j'}\frac{\dot{x}_{ij,\parallel}^m - \dot{p}_{ij,\parallel}^m}{\sqrt{\left\|\dot{x}_{ij,\parallel}^m - \dot{p}_{ij,\parallel}^m\right\|^2 + \epsilon}}, \tag{11}$$

with μ being the frictional coefficient. $f_{ij,\parallel}^m$ can be equivalently expressed in the following conservative form:

$$D_{ij}^m(d,\theta^{t+1},\theta^t,u,\omega) \triangleq \mu\Delta t\mathcal{A}_\epsilon(f_{ij,\perp}^m(d,\theta^t))\sqrt{\left\|\dot{x}_{ij,\parallel}^m - \dot{p}_{ij,\parallel}^m\right\|^2 + \epsilon}, \tag{12}$$

where we define $\mathcal{A}_\epsilon(x) \triangleq \sqrt{\|x\|^2+\epsilon} - \sqrt{\epsilon}$ and introduce a small constant ϵ to enforce smoothness as proposed in [37], which suffices for our co-design applications. Smoothing function for more accurate static-sliding mode switching [47] can also be used to replace the function $\sqrt{\|\bullet\|^2+\epsilon}$ in our formulation, which follows an identical differentiability analysis as in our Section VIII. We denote $U_{f\star}^{ij,i'j'} \triangleq \sum_m [D_{ij}^m + D_{i'j'}^m]$ as the frictional-damping energy, which allows us to integrate the frictional force seamlessly into the unconstrained optimization framework. Notably, we rely on the magnitude of the normal force and the normal vector of the separating plane from the tth timestep to predict the frictional force at the $t+1$th timestep as proposed in [47]. Although a more advanced approach in [62] has been proposed to iteratively estimate the normal and frictional forces at the $t+1$th timestep, the convergence of such iteration is not well-studied, complicating our subsequent differentiability analysis. Our evaluation shows that using the separating plane from the tth timestep is sufficiently accurate when operating with a small timestep size Δt.

In a similar manner to the normal contact forces, our frictional-damping energy remains under-determined due to the involvement of the separating plane's velocity parameters, u and ω. To resolve these parameters, we invoke the Maximal Dissipation Principle (MDP), which asserts that an appropriate frictional force should maximize the dissipation of tangential velocity. Since our frictional-damping energy quantifies the magnitude of tangential relative velocity, the MDP immediately implies that u and ω correspond to local minima of the frictional-damping energy. In light of these considerations, we define the following well-determined frictional-damping energy for a pair of convex hulls, H_{ij} and $H_{i'j'}$:

$$U_f^{ij,i'j'}(d,\theta^{t+1},\theta^t) \triangleq \min_{u,\omega} U_{f\star}^{ij,i'j'}(d,\theta^{t+1},\theta^t,u,\omega). \tag{13}$$

It can be shown that our frictional potential is well-defined, twice-differentiable, and imposes opposing forces on the two convex polyhedrons (see Section VIII). Unfortunately, our formulation does not impose opposing torques on the two convex polyhedrons. Instead, it only preserves the in-plane components of the torque. This is a an issue inherited from the potential-energy-based contact models such as [47], where the two contacting objects are not exactly touching each other, but a small margin is left in between. With these elements in place, we can now present the comprehensive formulation of the SDRS simulation function f. This function is structured as an unconstrained optimization, and it is expressed as follows:

$$\begin{aligned}\theta^{t+1} \triangleq \operatorname*{argmin}_{\theta^{t+1}} \mathcal{O}(d,\theta^{t+1},\theta^t,\theta^{t-1},u^t) \triangleq \sum_{ij} I_{ij}(d,\theta^{t+1},\theta^t,\theta^{t-1})+ \\ U_{\text{PD}}(\theta^{t+1},\theta^t,u^t) + \mu\sum_{ij,i'j'}^{i\neq i'}\left[U_c^{ij,i'j'}(d,\theta^{t+1}) + U_f^{ij,i'j'}(d,\theta^{t+1},\theta^t)\right].\end{aligned} \tag{14}$$

Notably, we require $i \neq i'$, i.e., we exclusively examine collisions between convex hulls originating from distinct robot links. This allows for each robot link to be represented by the (potentially overlapping) combination of convex hulls.

D. Forward Simulation & Backward Derivative Evaluation

To tackle the optimization problem represented by Equation 14 for forward simulation, we employ a conventional Newton's method paired with a line-search globalization technique. This approach guarantees that the solution consistently reduces the objective function, thereby ensuring collision-free robot states throughout the simulation. For each timestep in the simulation, we initiate the optimization process by setting $\theta^{t+1} \leftarrow \theta^t$. That said, we emphasize that our method can only guarantee that the simulator is collision-free at discrete time instances $\theta^t, \theta^{t+1}, \cdots$, but we cannot ensure that the robot motion is collision-free during the motion between θ^t, and

θ^{t+1}, which requires a continuous collision checker coupled with the line-search scheme. Unfortunately, prior works [63, 64] on Continuous Collision Detection (CCD) can only be adapted to linear or rigid motions, which are incapable of handling more general articulated motions owing to our nonlinear forward kinematics function $T_i(d, \theta^t)$. The recently proposed affine body dynamics [53] can be incorporated to truly achieve continuous-time collision-free properties. However, this method can significantly complicate our implementation so we choose not to incorporate it. In practice, we have never observed any tunneling artifact due to our small timestep. After the forward pass, we evaluate the derivatives using the adjoint method. In our Section IX, we show that the entire function $\mathcal{O}$ is globally twice-differentiable. As a result, it has well-defined derivatives by the implicit function theorem, under sufficiently small timestep sizes.

V. Evaluation

Substep	Percentage
$U_c^{ij,i'j'}$	30.51%
$U_f^{ij,i'j'}$	36.89%
Backward	30.85%
Other Steps	1.75%

TABLE II: Time breakdown per-iteration on average.

We have implemented our method using C++ with Python interface, where we use OpenMP to parallelize the update of bounding volume hierarchy and evaluation of the contact potential terms $U_{c,f}^{ij,i'j'}$. During the evaluation of each term, we use Newton's method to solve the internal small-scale optimization with respect to p and (u, ω) for $U_c^{ij,i'j'}$ and $U_f^{ij,i'j'}$, respectively. These internal optimizations are solved until the norm of the gradient ($\|\nabla_p U_{c\star}^{ij,i'j'}\|$ or $\|\nabla_{u,\omega} U_{f\star}^{ij,i'j'}\|$) is less than 10^{-8} before the energy values and derivatives can be evaluated using the implicit function theorem. We further warm-start these internal optimizations using the solution from the previous iteration, as long as these initial guesses do not lead to undefined values of $U_c^{ij,i'j'}$, in which case we recompute the distance between convex hulls using GJK and initialize the separating plane p from the middle surface. For the SDRS optimization (Equation 14), we use explicit eigen decomposition to factorize the Hessian matrix of $\mathcal{O}$ and we adjust the eigenvalues to be larger than 10^{-3} to ensure descendant directions are computed. Thanks to the use of reduced coordinates, the $|\theta| \times |\theta|$ Hessian matrix is rather small, making direct eigen decomposition possible. We terminate the optimization when $\|\nabla \mathcal{O}\|_\infty < 10^{-4}(N)$. Note that $\nabla \mathcal{O}$ has the same unit as force, so our termination condition implies that at most $10^{-4}N$ of force is left unconsidered and not converted into acceleration. For solving the co-design problem defined in Section III-A, we use the adaptive moment estimation (Adam) method [65]. We incorporate our simulator as an operator in the PyTorch framework [66] and we use a highly optimized routine for evaluating the derivatives of our simulation function using the adjoint method. During each co-design optimization, we first compute the derivative $\nabla_{d,c} L(\theta^H)$ and then update (d, c) via:

$$\begin{aligned} \underset{\substack{c+\Delta c \in \mathcal{C} \\ d+\Delta d \in \mathcal{D}}}{\operatorname{argmin}} \; & L(\theta^H) + \Delta d^T \nabla_d L(\theta^H) + \Delta c^T \nabla_c L(\theta^H) \\ \text{s.t.} \quad & \|\Delta d\| \leq \bar{\alpha}_d \wedge \|\Delta c\| \leq \bar{\alpha}_c, \end{aligned} \tag{15}$$

where $\bar{\alpha_d}$ and $\bar{\alpha_c}$ are hyper-parameters defining the learning rate for robot design and controller. The above optimization is convex and we solve it efficiently using CVX [67]. All the experiments are performed on a desktop machine with Intel Core i9-13900K CPU and 128GB memory.

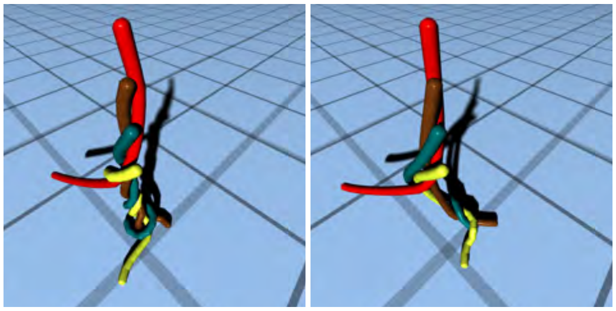

Fig. 6: Simulated frames of four rotating chains using mesh-based (left) and our convex-polyhedron-based shape representations (right), where we use [47] as the potential energy for mesh-based representation.

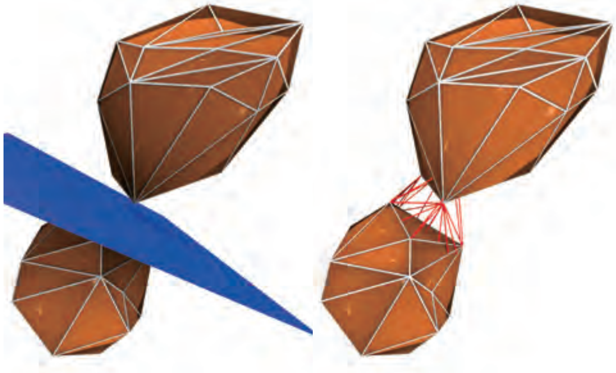

Fig. 7: We illustrate the two settings for one example. Our method uses a separating plane (blue) to formulate contact potential under a convex-polyhedron-based representation, while under a mesh-based representation, contact potential is formulated via distance between geometric primitives (red).

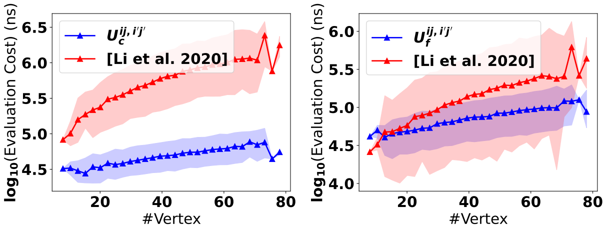

Fig. 8: We compare the evaluation cost of the contact potential ($U_c^{ij,i'j'}$) and the frictional damping potential ($U_f^{ij,i'j'}$) in these two cases.

A. Performance Evaluation

We evaluate our method in a row of benchmark problems. In Figure 6, we evaluate the computational cost. We simulate four rotating chains causing a large number of (self-)intersections. Each chain has # ball joints each having 2 degrees of freedom, totalizing $|\theta| = 4 \times 2 \times \#$ degrees of freedom. The simulation is conducted using two methods, with mesh-based and our shape representations based on convex polyhedrons. Under mesh-based representation, we use the (frictional) contact potentials proposed in [47] to replace our $U_{c,f}^{ij,i'j'}$. The per-timestep cost of both methods are plotted against # in Figure 9. The performance breakdown is summarized in Table II. Our major bottleneck lies in the valuation of potential computation and the backward differentiation. Thanks to the use of minimal coordinates, the dimension of our configuration space is rather low and the system matrix inverse in the Newton's step is fast to compute, whose overhead is neglectable (1.75% in Table II). Our contact potential is faster to evaluate than [47] as shown in Figure 8, although our potential takes a more complex form as illustrated in Figure 7. This is because, for two convex hulls each with M vertices, our potential $U_c^{ij,i'j'}$ only involves $O(2M)$ terms, while $O(M^2)$ terms are needed in the worst case under a mesh-based representation, one for each pair of geometric primitives. We generate random convex hulls with different number of vertices and plot the cost of computing $U_c^{ij,i'j'}$ in Figure 7. Our potential evaluation can be $2.55 - 32$ times faster than one using mesh-based representation.

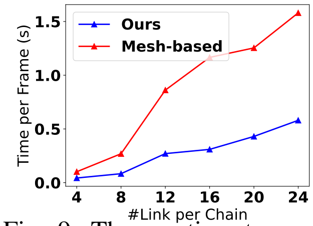

Fig. 9: The per-timestep cost of both methods are plotted against the number of links in each chain.

B. Comprehensive Derivative Information

In Figure 10, we show that SDRS can provide derivatives with respect to various design and control variables. We initialize three 8-link chains, where only the left (gray) chain is actuated and the two right chains are passive. We use an 1D translational actuator for the horizontal base position of the left chain. The goal of control is to have the left chain reach the (red) target point. We parameterize the control signal using a cubic spline with its 4 coefficients being the controller parameters, with an entire horizon of $H = 250$. In addition, we include the attachment points of each link on the left chain as additional 8 design variables, while relying on the shared vertices x_{ijk} to keep the chain connected. As shown in Figure 10 (right), our co-design can significantly outperform pure controller optimization by elongating the chain to reach the target point, taking 16.63 (min) to converge. Indeed, our co-design can further decrease the loss function by 98.5% and our optimized chain reaches the red target with an error around 0.2 times the link length.

C. Benchmark: Quadruped Locomotion

We illustrate our first benchmark in Figure 11, where we optimize the motion of a 9-link spider robot with 16 degrees of freedom walking on a 15°-sloped ground. We use a set of parameters $\bar{\alpha_c} = 3e^{-1}, \bar{\alpha_d} = 4e^{-3}$, and $H = 550$. The goal of optimization is to have its center-of-mass reach a target position by setting the loss function as $R(\theta^H) = -\|\text{COM}(\theta^H) - (10\cos(15), 0, 10\sin(15))\|^2$, where the COM is initially at $(0, 0, 0)$ and $-z$ is the gravitational direction. Our controller is parameterized in a similar way as in [11], by setting the PD-controller's position target $\theta_\star^t$ to be a linear combination of 4 sine waves with learnable magnitudes, phase shifts, and frequency. We further set the derivative target $\dot{\theta}_\star^t$ to be the time-derivative of these sine waves. We set the PD gains to be $k_p = 1e^2$ and $k_d = 1e^1$. During the co-optimization stage, we focus solely on optimizing the shape of the robot's 4 feet. Each robot foot has an initial 96 vertices, and using V-HACD, we represent each foot using 3 convex polyhedrons. The convergence history of the optimization with and without shape co-optimization is plotted in Figure 11, both taking 4 (min) to converge, where the version with shape co-optimization further increases the spider's walking distance by 8.78% from 4.90 to 5.33. Our improvement in this benchmark is due to the shape co-optimization that deforms the robot foot to have a flat and larger contact surface (Figure 11 bottom right), leading to larger tangential friction force that propels the robot to move forward.

We further compare our approach with Xu et al. [9] on this benchmark, which assumed that the robot shape is controlled by a cage mesh. As they use mesh-based (MB) simulator, we use IPC [47] to handle contact and friction. We use same task settings and simulator configuration as in our approach, to ensure that both simulators will generate nearly identical trajectories under the same PD signal and exhibit nearly identical performance when only controllers are optimized. For fairness, we tune the resolution of the cage used in Xu et al. [9] to have a same number of vertices as those in our method. The convergence history of the optimization with and without shape co-optimization with fine-cage mesh-based method is plotted in Figure 11, where the version with shape co-optimization further increase the spider's walking distance by 8.16% from 4.90 to 5.30. We observed that the improvements achieved by mesh-based methods are comparable to those of our approach, which is expected since this benchmark does not demand significant shape changes. However, a notable difference emerges in the optimized leg shape. As shown in Figure 11, both methods attempt to expand the spider's end effector to increase the contact surface. However, the mesh-based technique introduces additional notches and wedges that are less suitable for fabrication. In contrast, our approach enables convenient adjustment of local details by moving vertices in or out of the convex polyhedra, effectively resolving these artifacts.

D. Benchmark: Bipedal Locomotion

In Figure 12 and Figure 13, we optimize the motion of a 7-link bipedal robot with 12 degrees of freedom walking on the ground. In this benchmark, the initial shape of the robot's foot are not well-designed. They are relatively short in length and have smooth contact surfaces with the ground.

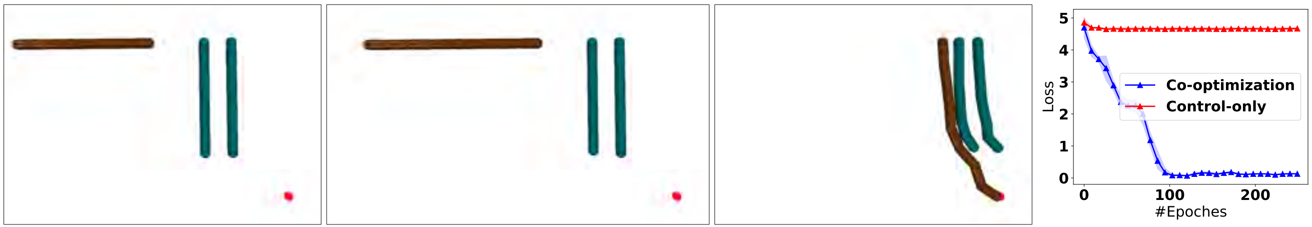

Fig. 10: We optimize the attachment points and link-lengths of a 8-link chain to reach a target red point (bottom right). From left to right: The initial configuration of two chains; the optimized chain with elongated links; the simulated sequence of the chain reaching the target red point; and the convergence history of the benchmark with and without co-design.

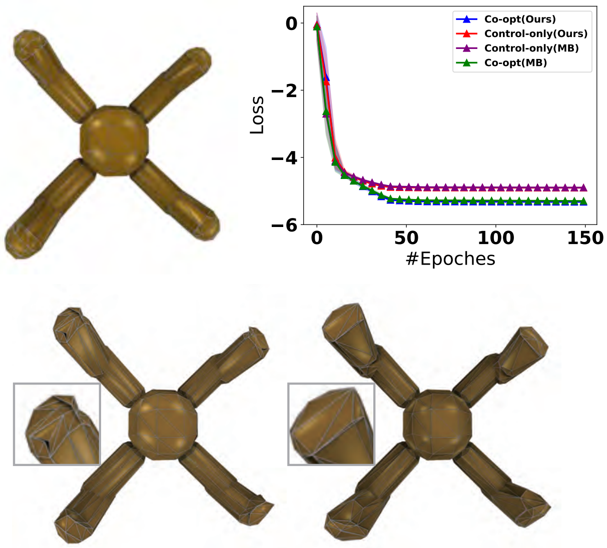

Fig. 11: Top row: We show the original spider leg shape (left) and the convergence history (right). Bottom row: We show optimized leg shape with our approach (left) and mesh-based (MB) method (right). The shape of end-effector is highlighted to the left, which shows that mesh-based method creates unnecessary notches and wedges.

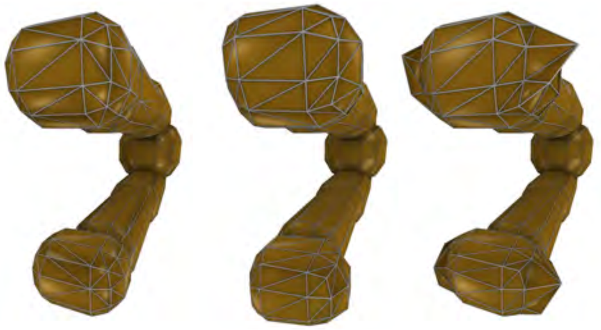

Fig. 12: From left to right, we show the original bipedal foot shape, optimized foot shape using mesh-based methods with coarse cages and fine cages.

Therefore, solely optimizing the controller is insufficient to maintain passive balance for the robot during walking. We use a different set of parameters $R(\theta^H) = -\|\text{COM}(\theta^H) - (0,3,0)\|^2, H = 400, \bar{\alpha_c} = 1e^{-2}$ and $\bar{\alpha_d} = 4e^{-3}$, where the COM is initial at $(0,0,0)$ and $-z$ is the gravitational direction. We set the PD gains to be $k_p = 1e^3$ and $k_d = 1e^2$.

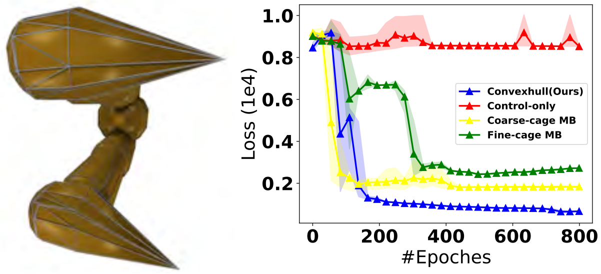

Fig. 13: We show optimized bipedal foot shape with our approach (left) and the convergence history (right).

During co-optimization, we confine our design space to the shape of the robot's 2 feet, along with controller parameters. Each robot foot has 96 vertices. Using V-HACD, we represent each foot using 3 convex polyhedrons. We compare our method with Xu et al. [9], where we experiment with two resolutions of an axis-aligned case. According to their original paper, we first experimented with a coarse cage having 26 vertices. For fairness, we also increase the cage resolution to have 96 vertices, matching the degree of freedom used in our method. As illustrated in Figure 12, the coarse cage cannot effectively deform the feet shape to maintain balance, while the fine cage can slightly deform the feet. Instead, our approach in Figure 13 significantly deforms the robot's feet, creating rough protrusions, increasing the length of the feet, and generating flat contact with the ground, enabling the robot to easily maintain balance even when walking with an open-loop controller. The convergence history of co-optimization using various methods are plotted in Figure 13 (right).

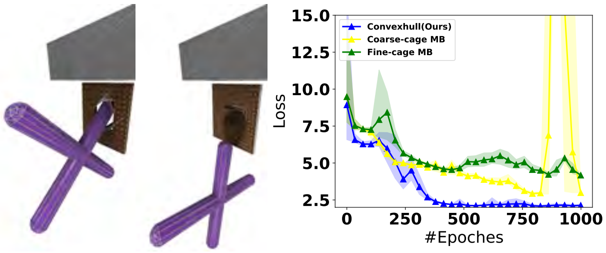

Fig. 14: Our benchmark requires a single-jaw gripper to grasp an X-shaped object. Our co-optimization (left) change the gripper's topology by creating a hole to secure the object, while the mesh-based method (middle) failed to grasp the object. We show the convergence history of the benchmark using our approach and mesh-based method (right).

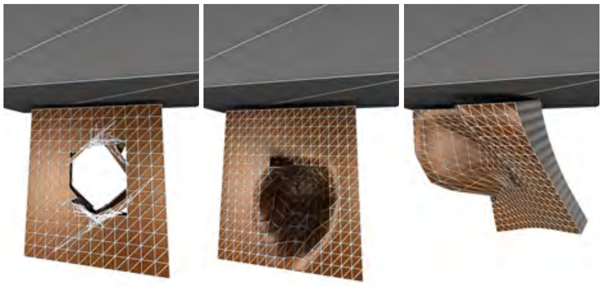

Fig. 15: A comparison of the two techniques in optimizing the single-jaw gripper. Our co-optimization (left) change the topology of original gripper shape, creating a hole to form a firm grasp of the object. Co-optimization by mesh-based method using fine cages (middle) generate local concavity. Coarse cages (right) exhibit global deformation and fail to generate useful local deformations.

E. Benchmark: Single-Jaw Gripper

We highlight the capability of our design space to handle topological changes. In Figure 14, we optimize the motion and shape of a single-jaw gripper to grasp an X-shaped object placed on the ground. The gripper has 1 link with 6 degrees of freedom and the object also has 6 degrees of freedom. We set $H = 480, \bar{\alpha_c} = 5e^{-2}$ and $\bar{\alpha_d} = 2e^{-3}$. We divide the grasping process into four stages. The first stage involves lowering the gripper to the appropriate position. In the second stage, the gripper approaches and grasps the object. The third stage involves adjusting the gripper to its initial orientation. Finally, in the fourth stage, the gripper is raised to a target position. During the first two stages, we define a penalty function $L(\theta^H) = \|COM_{\text{object}}(\theta^H)\|^2$ with the object's center of mass initialized at zero, aiming to keep the object staying still when the gripper approaches and from a firm grasp. During the last two stages, we penalize any relative motions between the object and the gripper: $L(\theta^H) = \|COM_{\text{object}}(\theta^H) - COM_{\text{gripper}}(\theta^H)\|^2$. Clearly, regardless of individual controller optimization efforts, the initial gripper will fail to grasp the object successfully due to the lack of a well-designed mechanism to secure the object, meaning that co-optimization is necessary. We extensively compared the performance of our approach against the mesh-based method [9] using coarse and fine cages in Figure 15. The initial gripper has 1538 vertices uniformly distributed. Our coarse cages use only 98 vertices, and as usual, we use 1538 vertices for the fine cage for fairness of comparison. We find that the mesh-based method using fine cages can optimize a groove on the gripper that could be used to secure the object. However, due to the low friction coefficient, the object still cannot be successfully grasped. Instead, the mesh-based method using coarse cages struggles to generate large deformations conducive to grasping the object. Finally, none of these mesh-based methods can induce topological changes to help solve this task. In contrast, our method allows for the optimization of the gripper's topology. We evenly divide the gripper into 156 convex blocks, which brings a vertex count similar to that of the mesh-based method. Our method carefully drills a hole on the gripper for one handle in the X-shaped object to pass through and be locked in place, enabling successful object retrieval even with a lower friction coefficient. The convergence history is plotted in Figure 14, where our method takes around 55 (min) to converge.

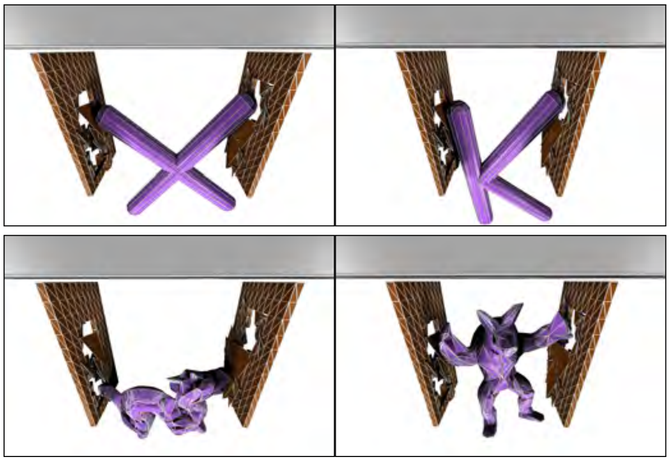

Fig. 16: The results of successfully grasping 4 different objects with a parallel-jaw gripper optimized using our approach, inducing both topological and geometric shape change.

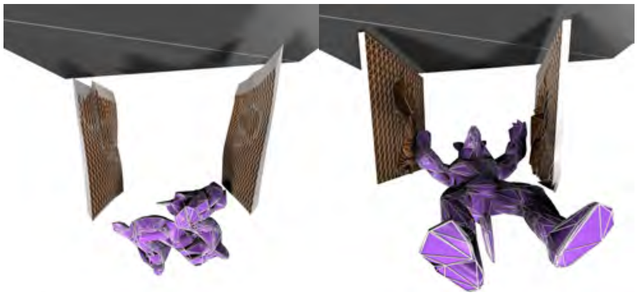

Fig. 17: The grippers optimized with mesh-based methods using coarse cages (left) and fine cages (right) both fail to grasp certain objects.

F. Benchmark: Universal Parallel-Jaw Gripper

In Figure 16, we use our method to optimize the shape of a parallel-jaw gripper to grasp a set of four different objects and reach the target position located in random orientation. The gripper has 3 links and 5 degrees of freedom, and for this benchmark, we use a preset parameterization of the controller. Specifically, we fix the pose of the closed gripper, denoted as $\theta_\star$, and we set all the position target $\theta_\star^t = \theta_\star$ throughout the trajectory. Our design goal is to have the gripper firmly grasp each objects by penalizing any relative motions between the ball and the gripper: $L(\theta^H) = \|COM_{\text{object}}(\theta^H) - COM_{\text{gripper}}(\theta^H)\|^2$. In each iteration, the gripper sequentially grasps 4 objects and reaches a randomly assigned target position, and we optimize these 4 trajectories collectively. Due to a low friction coefficient, none of these objects can be grasp successfully using initial gripper shape. We run iterations of Adam with a horizon of $H = 180$ for each trajectory and the learning rate $\bar{\alpha_d} = 5e^{-3}$. When using our approach, we decompose each jaw into 156 convex blocks. We find our optimization leads to both geometry and topology changes of the gripper. These changes, allow the complex shapes of all 4 objects to fit tightly into the dents of the gripper and form firm grasps. Meanwhile, mesh-based methods using neither coarse cages nor fine cages can grasp all these objects, the failure cases are shown in Figure 17 and the convergence history is summarized in Figure 18.

G. Benchmark: Three-Fingered Hand

In Figure 19, We optimized the shape of a three-fingered robot gripper to grasp 3 different objects and reach the target position located in any orientation. The gripper has 6 links and 6 degrees of freedom. Using V-HACD, we represent each link using 4 convex polyhedrons. In each iteration, the gripper sequentially grasps 3 objects and reaches a randomly assigned target position, while we optimize these 3 trajectories collectively. We run iterations of Adam with the horizon $H = 200$ for each trajectory and the learning rate $\bar{\alpha_d} = 1e^{-3}$. Specifically, we fix the pose of the closed gripper, denoted as $\theta_\star$, and we set all the position target $\theta_\star^t = \theta_\star$ throughout the trajectory with PD gains being $k_p = 3e^2$ and $k_d = 3e^0$. For each target position, we then move the base of the gripper to that target position, so there are no control parameters. Our design goal is to have the gripper firmly grasp each objects by penalizing any relative motions between the ball and the gripper: $L(\theta^H) = \|COM_{\text{ball}}(\theta^H) - COM_{\text{gripper}}(\theta^H)\|^2$. We compare our approach with mesh-based method with fine cages. The convergence history of this benchmark is summarized in Figure 19. We found that both methods performed well in this benchmark and our approach takes around 500 iterations to converge, while mesh-based method takes 400 iterations. Again, this is not surprising because the optimized gripper shape does not require topological changes. However, the optimized shape using mesh-based method has some thin structures that are less suitable for manufacturing. This is because under a fixed topology, thin structures can be easily created when the optimizer moves neighboring vertices close to each other. Instead, by maintaining the convexity of geometric primitives, our method creates bulky lumps on the robot links that can fit the geometry of the object to form a firm grasp, while being manufacture-friendly.

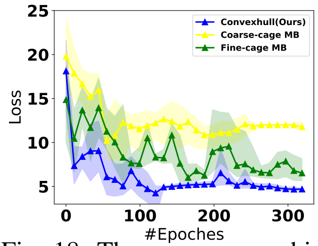

Fig. 18: The convergence history of optimizing the parallel-jaw gripper using various methods.

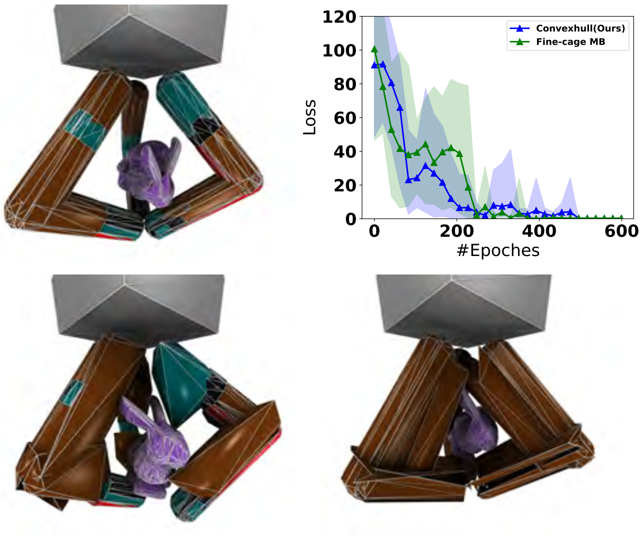

Fig. 19: We optimize the three-fingered gripper to grasp three objects. We visualize the optimial gripper shape (top left), the optimized shape using our method (bottom left) and mesh-based method (bottom right). The convergence history of both methods are illustrated in the top right corner.

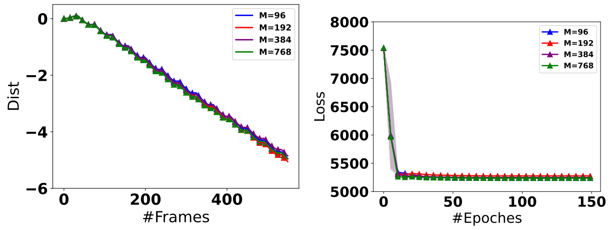

Fig. 20: We compare the dynamics (left) and the policy convergency (right) under different number of vertices.

H. Ablation: Impact of Polyhedron Resolution

We investigate whether the choice of the number of polyhedral vertices affects the performance of the simulator and the gradient-based policy optimization. Experiments are conducted on the quadruped locomotion benchmark. For the spider robot, we uniformly sample the number of points to discretize each joint shape, resulting in four settings with $M = 96, 192, 384, 768$, respectively. Note that we still keep a same total mass but evenly distribute the mass on the vertices. In Figure 20 (left), we have the spider robot move forward under the same PD control signals and plot the moving distance against the simulation frame. We observe the four curves almost overlap with the final travel distance in range -4.890 ± 0.097m. In Figure 20 (right), we test the impact of different M on the policy convergence. Using the same optimization settings as in the quadruped locomotion benchmark, we record the convergence history under the four different M. Again, we observe that the convergence curves almost overlap, and the final loss at convergence has an error range of 5257 ± 20. These experiments demonstrate that the choice of the number of vertices has almost negligible influence.

I. Ablation: Impact of Mass Distribution

To achieve fully differentiable simulator, we have assumed that the mass of the robot link is evenly distributed among the convex hull vertices. We investigate the impact of this assumption by comparing uniformly distributed and vertex-concentrated mass matrices. We conduct experiments on both quadruped spider and bipedal robots. We simulate the two robots with our optimized locomotion controller under the two mass distributions. In Figure 21, we plot the robot moving distance against the simulation frame, and we observe the plots under the two mass distributions almost overlap. The final moving distance under the two mass distributions are -5.063m and -5.053m for the quadruped spider robot with an error of 0.19%. The moving distance for the bipedal robot are -7.666m and -7.762m, with an error of 1.23%.

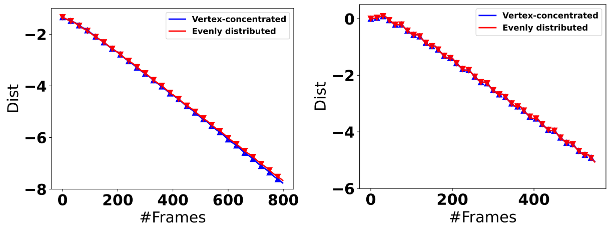

Fig. 21: We compare the dynamics under our vertex-concentrated and evenly distribution with bipedal (left) and quadruped (right) locomotion benchmark.

J. Comparison with MuJoCo

We conduct a series of comparative experiments with MuJoCo [55] to demonstrate the physical accuracy of our simulator and to validate the sim-to-real performance of our co-design approach. In the single-jaw gripper benchmark, we use the same settings in SDRS and MuJoCo where we adopt our optimizer grasping controller. In both simulators, the gripper successfully grasps the object. The the object's center-of-mass altitude is plotted in Figure 22 (left) against the simulation frame, where curves generated by the two simulators almost overlap. Meanwhile, we also tested the performance of the shape and locomotion policy of the quadruped robot and bipedal robot, optimized using SDRS while tested in MuJoCo. In MuJoCo, the bipedal robot was still able to maintain balance, and both robots were able to perform forward locomotion using the same policy. The moving distance is plotted in Figure 22 against the simulation frames, where the curves under the two simulators are highly consistent.

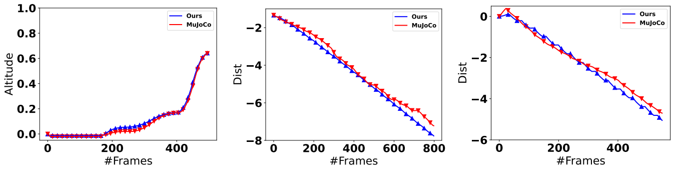

Fig. 22: The object's center-of-mass altitude plotted against the simulation frame on gripper benchmark (left), and moving distance against the simulation frame on bipedal (middle) and quadruped (right) locomotion benchmark.

Additionally, we test the performance of our optimized design under a different, gradient-free controller optimization algorithm. For this experiment, we use two benchmarks: the bipedal locomotion and the single-jaw gripper. For each benchmark, we setup the original robot design and our optimized design in MuJoCo. We then use Soft-Actor-Critic reinforcement learning algorithm [68] to optimize the controller. We plot the cumulative reward against the training epoch for both benchmarks in Figure 23. We observe that in both benchmarks, the designs optimized using our method consistently lead to better policy performance.

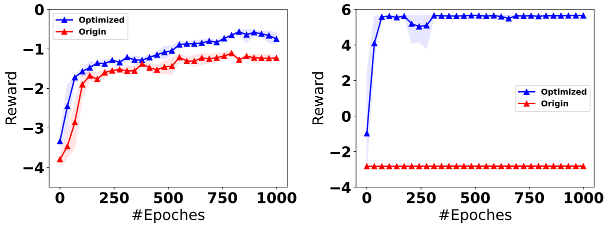

Fig. 23: The convergence history of RL under the original design and our optimized design on the bipedal locomotion benchmark (left) and the single-jaw gripper benchmark (right).

We further validated the physical accuracy of our contact and friction model through two additional experiments. In our first experiment, a fixed wall and two movable objects were placed on a flat plane. By applying an initial angular velocity to the second object, it collides with the first object, and due to the effect of friction, the two objects eventually form a stable angle with the wall, as shown in Figure 24 (top left). We performed this forward simulation in both SDRS and MuJoCo, recording the vertical angles of the two objects, *obj1* and *obj2*, in radians over the simulation time. We observed that the angular variation curves of the two objects in both simulators were nearly identical, demonstrating that our friction model is physically accurate. In our second experiment, we validated the book stacking problem [69], where 20 planks are stacked sequentially, each extending outward to the maximum theoretical distance from the bottom plank without collapsing. Based on the theoretical maximum distance, the planks were placed, and to account for simulation errors, they were shifted inward by ϵ, where ϵ is 0.1% of the plank's length. In both SDRS and MuJoCo, the planks did not collapse and maintained for at least 60 seconds of simulation time. Figure 24 (bottom right) records the resultant force on the right side of each plank, showing that the results in SDRS are completely consistent with those in MuJoCo.

VI. Conclusion & Discussion

We introduce SDRS, a novel robot simulator that features provable differentiability with respect to state, control, and robot shape parameters, accommodating large geometric and topological changes of robot link shapes. There are several avenues for enhancing our differentiable simulator. Firstly, our current contact potential energy model, which requires solving a small optimization problem at each evaluation, leads to relatively high computational demands compared to mesh-based contact potential models, even with the use of warm-start initial guesses. In the future, we plan to leverage parallel processors, such as GPUs, to accelerate this process. Secondly, our simulator does not guarantee collision-free states throughout continuous articulated motions. While we have not encountered this issue in our experiments, future research incorporating CCD for articulated motions will be crucial for ensuring robust simulations. Further, to circumvent the

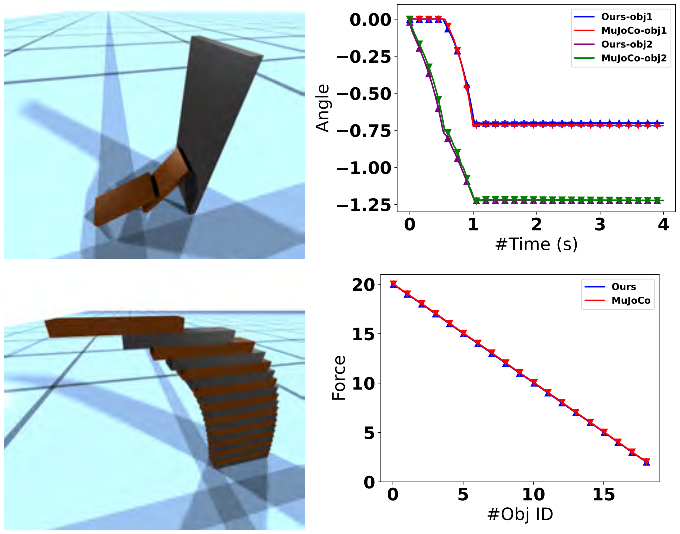

Fig. 24: In the first experiment, the two objects eventually form a certain angle with the wall (top left), along with the curves of their angles changing over simulation time (top right). In the second experiment, planks are stacked at the maximum extension distance without collapsing (bottom left), along with force distribution on the right side of each plank (bottom right).

need for differentiating integrals over convex hulls in I_{ij}, we currently assume that the robot's mass is concentrated at the vertices of the convex hulls, which may lead to large discretization error. Moving forward, we plan to develop more precise formulas for the exact derivatives of I_{ij}. Lastly, our method relies solely on the gradient-based optimization and cannot design discrete robot parameters such as joint structure, which requires sampling-based optimization algorithms as done in [70].

Acknowledgment

This project was partially funded by Meta. It was also partially funded by the Research Grants Council of Hong Kong (Ref: 17210222), and by the Innovation and Technology Commission of the HKSAR Government under the ITSP-Platform grant (Ref: ITS/335/23FP) and the InnoHK initiative (TransGP project). Part of the research was conducted in the JC STEM Lab of Robotics for Soft Materials, funded by The Hong Kong Jockey Club Charities Trust.

References

[1] G. C. Causey and R. D. Quinn, “Gripper design guidelines for modular manufacturing,” in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, IEEE, vol. 2, 1998, pp. 1453–1458.

[2] J. Burdick and P. Fiorini, “Minimalist jumping robots for celestial exploration,” *The International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 653–674, 2003.

[3] C. Hubicki et al., “Atrias,” *Int. J. Rob. Res.*, vol. 35, no. 12, 1497–1521, Oct. 2016.

[4] I. Poulakakis and J. W. Grizzle, “The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper,” *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1779–1793, 2009.

[5] S. Kajita et al., “Biped walking stabilization based on linear inverted pendulum tracking,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2010, pp. 4489–4496.

[6] S. Ha, S. Coros, A. Alspach, J. Kim, and K. Yamane, “Computational co-optimization of design parameters and motion trajectories for robotic systems,” *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1521–1536, 2018.

[7] A. Zhao et al., “Robogrammar: Graph grammar for terrain-optimized robot design,” *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–16, 2020.

[8] Y. Kim, Z. Pan, and K. Hauser, “Mo-bbo: Multi-objective bilevel bayesian optimization for robot and behavior co-design,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 9877–9883.

[9] J. Xu et al., “An End-to-End Differentiable Framework for Contact-Aware Robot Design,” in *Proceedings of Robotics: Science and Systems*, Virtual, Jul. 2021.

[10] T. Liao et al., “Data-efficient learning of morphology and controller for a microrobot,” in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 2488–2494.

[11] Y. Hu et al., “Chainqueen: A real-time differentiable physical simulator for soft robotics,” in *2019 International conference on robotics and automation (ICRA)*, IEEE, 2019, pp. 6265–6271.

[12] S. Chen, X. Ma, and Z. Xu, “Imitation learning via differentiable physics,” *arXiv preprint arXiv:2206.04873*, 2022.

[13] C. Song and A. Boularias, “Learning to slide unknown objects with differentiable physics simulations,” *arXiv preprint arXiv:2005.05456*, 2020.

[14] B. Deng, K. Genova, S. Yazdani, S. Bouaziz, G. Hinton, and A. Tagliasacchi, “Cvxnet: Learnable convex decomposition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 31–44.

[15] W. Hönig, J. A. Preiss, T. S. Kumar, G. S. Sukhatme, and N. Ayanian, “Trajectory planning for quadrotor swarms,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 856–869, 2018.

[16] V. Megaro, B. Thomaszewski, M. Nitti, O. Hilliges, M. Gross, and S. Coros, “Interactive design of 3d-printable robotic creatures,” *ACM Trans. Graph.*, vol. 34, no. 6, Nov. 2015.

[17] R. Zhang, T. Auzinger, and B. Bickel, “Computational design of planar multistable compliant structures,” *ACM Trans. Graph.*, vol. 40, no. 5, Oct. 2021.

[18] G. Maloisel, C. Schumacher, E. Knoop, R. Grandia, and M. Bächer, “Optimal design of robotic character kinematics,” *ACM Transactions on Graphics (TOG)*, vol. 42, no. 6, pp. 1–15, 2023.

[19] Y. Wang et al., “Curriculum-based co-design of morphology and control of voxel-based soft robots,” in *The Eleventh International Conference on Learning Representations*, 2022.

[20] G. Fadini, T. Flayols, A. Del Prete, and P. Souères, “Simulation aided co-design for robust robot optimization,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 306–11 313, 2022.

[21] P. Ma et al., “Diffaqua: A differentiable computational design pipeline for soft underwater swimmers with shape interpolation,” *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, p. 132, 2021.

[22] T. Dinev, C. Mastalli, V. Ivan, S. Tonneau, and S. Vijayakumar, “A versatile co-design approach for dynamic legged robots,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2022, pp. 10 343–10 349.

[23] F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum, and J. Z. Kolter, “End-to-end differentiable physics for learning and control,” *Advances in neural information processing systems*, vol. 31, 2018.

[24] Y.-L. Qiao, J. Liang, V. Koltun, and M. C. Lin, “Efficient differentiable simulation of articulated bodies,” in *International Conference on Machine Learning*, PMLR, 2021, pp. 8661–8671.

[25] E. Heiden, M. Macklin, Y. S. Narang, D. Fox, A. Garg, and F. Ramos, “DiSECt: A Differentiable Simulation Engine for Autonomous Robotic Cutting,” in *Proceedings of Robotics: Science and Systems*, Virtual, Jul. 2021.

[26] T. Du et al., “Diffpd: Differentiable projective dynamics,” *ACM Trans. Graph.*, vol. 41, no. 2, Nov. 2021.

[27] T. Stuyck and H.-y. Chen, “Diffxpbd: Differentiable position-based simulation of compliant constraint dynamics,” *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 6, no. 3, pp. 1–14, 2023.

[28] Z. Huang et al., “Differentiable solver for time-dependent deformation problems with contact,” *ACM Transactions on Graphics*, vol. 43, no. 3, pp. 1–30, 2024.

[29] A. Gjoka, E. Knoop, M. Bächer, D. Zorin, and D. Panozzo, “Soft pneumatic actuator design using differentiable simulation,” in *ACM SIGGRAPH 2024 Conference Papers*, 2024, pp. 1–11.

[30] A. Spielberg, A. Zhao, Y. Hu, T. Du, W. Matusik, and D. Rus, “Learning-in-the-loop optimization: End-to-end control and co-design of soft robots through learned deep latent representations,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[31] T. Takahashi, J. Liang, Y.-L. Qiao, and M. C. Lin, “Differentiable fluids with solid coupling for learning and control,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 6138–6146.

[32] J. Liang, M. Lin, and V. Koltun, “Differentiable cloth simulation for inverse problems,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[33] H. Su, X. Li, T. Xue, C. Jiang, and M. Aanjaneya, “A generalized constitutive model for versatile mpm simulation and inverse learning with differentiable physics,” *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 6, no. 3, pp. 1–20, 2023.

[34] P. Ma, T. Du, J. B. Tenenbaum, W. Matusik, and C. Gan, “Risp: Rendering-invariant state predictor with differentiable simulation and rendering for cross-domain parameter estimation,” *arXiv preprint arXiv:2205.05678*, 2022.

[35] M. Geilinger, D. Hahn, J. Zehnder, M. Bächer, B. Thomaszewski, and S. Coros, “Add: Analytically differentiable dynamics for multi-body systems with frictional contact,” *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–15, 2020.

[36] K. Werling, D. Omens, J. Lee, I. Exarchos, and C. K. Liu, “Fast and feature-complete differentiable physics engine for articulated rigid bodies with contact constraints,” in *Robotics: Science and Systems*, 2021.

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

[38] T. Howell, S. Le Cleac’h, J. Bruedigam, Z. Kolter, M. Schwager, and Z. Manchester, “Dojo: A differentiable physics engine for robotics,” *arXiv preprint arXiv:2203.00806*, 2022.

[39] Z. Pan, B. Ren, and D. Manocha, “Gpu-based contact-aware trajectory optimization using a smooth force model,” in *Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA ’19, New York, NY, USA: Association for Computing Machinery, 2019.

[40] I. Mordatch, E. Todorov, and Z. Popović, “Discovery of complex behaviors through contact-invariant optimization,” *ACM Trans. Graph.*, vol. 31, no. 4, Jul. 2012.

[41] C. Hazard, N. Pollard, and S. Coros, “Automated design of robotic hands for in-hand manipulation tasks,” *International Journal of Humanoid Robotics*, vol. 17, no. 01, p. 1 950 029, 2020.

[42] R. Desai, B. Li, Y. Yuan, and S. Coros, “Interactive co-design of form and function for legged robots using the adjoint method,” *arXiv preprint arXiv:1801.00385*, 2018.

[43] E. Guendelman, R. Bridson, and R. Fedkiw, “Nonconvex rigid bodies with stacking,” *ACM transactions on graphics (TOG)*, vol. 22, no. 3, pp. 871–878, 2003.

[44] E. Catto, “Iterative dynamics with temporal coherence,” in *Game developer conference*, vol. 2, 2005.

[45] K. M. Jatavallabhula et al., “Gradsim: Differentiable simulation for system identification and visuomotor control,” *International Conference on Learning Representations (ICLR)*, 2021.

[46] D. Harmon, E. Vouga, B. Smith, R. Tamstorf, and E. Grinspun, “Asynchronous contact mechanics,” New York, NY, USA: ACM, 2009.

[47] M. Li et al., “Incremental potential contact: Intersection- and inversion-free large deformation dynamics,” *ACM Trans. Graph. (SIGGRAPH)*, vol. 39, no. 4, 2020.

[48] K. Tracy, T. A. Howell, and Z. Manchester, “Differentiable collision detection for a set of convex primitives,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 3663–3670.

[49] A. N. Iusem, “On the convergence properties of the projected gradient method for convex optimization,” *Computational & Applied Mathematics*, vol. 22, pp. 37–52, 2003.

[50] K. Mamou, E Lengyel, and A Peters, “Volumetric hierarchical approximate convex decomposition,” *Game engine gems*, vol. 3, pp. 141–158, 2016.

[51] Z. Pan and D. Manocha, “Time integrating articulated body dynamics using position-based collocation methods,” in *International Workshop on the Algorithmic Foundations of Robotics*, Springer, 2018, pp. 673–688.

[52] S. Martin, B. Thomaszewski, E. Grinspun, and M. Gross, “Example-based elastic materials,” in *ACM SIGGRAPH 2011 papers*, 2011, pp. 1–8.

[53] L. Lan, D. M. Kaufman, M. Li, C. Jiang, and Y. Yang, “Affine body dynamics: Fast, stable and intersection-free simulation of stiff materials,” *ACM Trans. Graph.*, vol. 41, no. 4, Jul. 2022.

[54] J. Tan, K. Liu, and G. Turk, “Stable proportional-derivative controllers,” *IEEE Computer Graphics and Applications*, vol. 31, no. 4, pp. 34–44, 2011.

[55] T. Erez, Y. Tassa, and E. Todorov, “Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx,” in *2015 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2015, pp. 4397–4404.

[56] J. Borwein and A. Lewis, *Convex Analysis*. Springer, 2006.

[57] S. Suthaharan and S. Suthaharan, “Support vector machine,” *Machine learning models and algorithms for big data classification: thinking with examples for effective learning*, pp. 207–235, 2016.

[58] P.-H. Chen, C.-J. Lin, and B. Schölkopf, “A tutorial on ν-support vector machines,” *Applied Stochastic Models in Business and Industry*, vol. 21, no. 2, pp. 111–136, 2005.

[59] F. E. Udwadia and G. Di Massa, “Sphere rolling on a moving surface: Application of the fundamental equation of constrained motion,” *Simulation Modelling Practice and Theory*, vol. 19, no. 4, pp. 1118–1138, 2011.

[60] F. E. Udwadia and N. Mogharabin, “The use of zero-mass particles in analytical and multi-body dynamics: Sphere rolling on an arbitrary surface,” *Journal of Applied Mechanics*, vol. 88, no. 12, p. 121 006, 2021.

[61] P. Lindemann, “The gilbert-johnson-keerthi distance algorithm,” *Algorithms in Media Informatics*, 2009.

[62] D. M. Kaufman, S. Sueda, D. L. James, and D. K. Pai, “Staggered projections for frictional contact in multibody systems,” in *ACM SIGGRAPH Asia 2008 papers*, 2008, pp. 1–11.

[63] M. Li, D. M. Kaufman, and C. Jiang, “Codimensional incremental potential contact,” *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–24, 2021.

[64] Z. Ferguson et al., “Intersection-free rigid body dynamics,” *ACM Transactions on Graphics*, vol. 40, no. 4, 2021.

[65] D. P. Kingma, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[66] A. Paszke et al., “Automatic differentiation in pytorch,” 2017.

[67] M. Grant, S. Boyd, and Y. Ye, “Cvx users’ guide,” *online: http://www. stanford. edu/boyd/software. html*, 2009.

[68] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *International conference on machine learning*, Pmlr, 2018, pp. 1861–1870.

[69] J. F. Hall, “Fun with stacking blocks,” *American journal of physics*, vol. 73, no. 12, pp. 1107–1116, 2005.

[70] Y. Yuan, Y. Song, Z. Luo, W. Sun, and K. M. Kitani, “Transform2act: Learning a transform-and-control policy for efficient agent design,” in *International Conference on Learning Representations*, 2022.

VII. Well-defined Contact Potential

In the analysis below, we always assume that $s \in (0,1)$. Under very mild assumptions, we show that our contact potential $U_c^{ij,i'j'}(d,\theta)$ is a well-defined, twice-differentiable function of its parameters θ and d. We further show that the potential induces opposing forces on the two convex hulls H_{ij} and $H_{i'j'}$. The main complication comes from the fact that $U_c^{ij,i'j'}$ relies on an internal optimization with respect to the separating plane, let us denote its set of locally optimal solution as:

$$p_\star = \left(n_\star, n_{0\star}\right) \in \operatorname*{argmin}_{p} U_{c\star}^{ij,i'j'}(d,\theta,p). \tag{16}$$

We begin by showing the following basic facts:

Lemma VII.1. *i) For any* $U_c^{ij,i'j'} < \infty$, $\|n\| \in (0,1)$*; ii)* $U_c^{ij,i'j'}$ *is a well-defined function of* d,θ*; iii)*

If dist$(H_{ij}, H_{i'j'}) \geq 2s/(1-s)$*, then* $U_c^{ij,i'j'} = 0$*; iv)* $\lim_{dist(H_{ij},H_{i'j'})\to 0} U_c^{ij,i'j'} = \infty$.

Proof. i) If $\|n\| \geq 1$, then the first term $P_s(1-\|n\|)$ in $U_{c\star}^{ij,i'j'}$ is undefined. If $\|n\| = 0$, then the remaining terms in $U_{c\star}^{ij,i'j'}$ reads:

$$\sum_{m=1}^{M} \left[P_s(-o) + P_s(o)\right]. \tag{17}$$

At least one of these terms are undefined, so $\|n\| \in (0,1)$.

ii) By construction, each term in $U_{c\star}^{ij,i'j'}$ is a convex function in p, so it has only one (global) minimum, proving that $U_c^{ij,i'j'}$ is single-valued, i.e., a well-defined function of d, θ.

iii) In this case, we must have two points $x_{ij} \in H_{ij}$ and $x_{i'j'} \in H_{i'j'}$ realizing the closest distance, i.e., $\text{dist}(x_{ij}, x_{i'j'}) \geq 2s/(1-s)$. Now, let us set p as the scaled, middle separating plane between x_{ij} and $x_{i'j'}$ by setting:

$$n = \frac{x_{i'j'} - x_{ij}}{\|x_{i'j'} + x_{ij}\|}(1-s) \text{ and } o = -n^T \frac{x_{i'j'} + x_{ij}}{2}(1-s). \tag{18}$$

Plugging this solution into $U_{c\star}^{ij,i'j'}$ and we have $U_{c\star}^{ij,i'j'} = 0$ by the local support of P_s. But by ii), $U_{c\star}^{ij,i'j'}(d,\theta,p)$ is convex in p and its minimal value can only be even smaller. But $U_{c\star}^{ij,i'j'} \geq 0$ by construction, so we conclude that $U_c^{ij,i'j'} = 0$ as desired.

iv) Given an arbitrary large number $M > 0$, there must be some $s_0 > 0$ such that $P_s(x) > M$ on $(0, s_0)$. We consider any two convex polyhedrons such that $\text{dist}(H_{ij}, H_{i'j'}) < s_0/(1-s_0)$. In this case, for any separating plane p, there must be some point, assumed to be $[T_{i'}(d,\theta)x_{i'j'}^m]$ without a loss of generality, such that:

$$0 < \frac{p^T T_{i'}(d,\theta) x_{i'j'}^m}{\|n\|} < \frac{s_0}{1-s_0}, \tag{19}$$

where we divide by $\|n\|$ to normalize the separating plane, leading to a standard point-to-plane distance. Now consider two cases. Case I: If $\|n\| > 1 - s_0$, then the first term $P_s(1-\|n\|) > M$, leading to $U_c^{ij,i'j'} > M$. Case II: If $\|n\| \leq 1 - s_0$, then we have:

$$0 < p^T T_{i'}(d,\theta) x_{i'j'}^m < s_0 \Rightarrow P_s(p^T T_{i'}(d,\theta) x_{i'j'}^m) > M, \tag{20}$$

again leading to $U_c^{ij,i'j'} > M$, from which the result follows. □

The above result already allows us to use BVH to quickly prune non-intersecting convex polyhedrons when their distance is larger than $2s/(1-s)$.

A. Local Differentiability

To proceed, we first prove differentiable properties when $U_c^{ij,i'j'} > 0$ and then extend these properties to all cases.

Lemma VII.2. *If* $U_c^{ij,i'j'} > 0$*, then i) at least one of the terms of form* $P_s(-p_\star^T T_i(d,\theta) x_{ij}^m)$ *or* $P_s(p_\star^T T_{i'}(d,\theta) x_{i'j'}^m)$ *is positive and ii)* $P_s(1 - \|n_\star\|) > 0$.

Proof. i) If all the terms of form $P_s(-p_\star^T T_i(d,\theta) x_{ij}^m) = 0$ and $P_s(p_\star^T T_{i'}(d,\theta) x_{i'j'}^m) = 0$, we must have:

$$\frac{\partial P_s(-p_\star^T T_i(d,\theta) x_{ij}^m)}{\partial n_\star} = \frac{\partial P_s(p_\star^T T_{i'}(d,\theta) x_{i'j'}^m)}{\partial n_\star} = 0. \tag{21}$$

Next, by Lemma VII.1 i), we know that $\|n\| \in (0,1)$. But $U_c^{ij,i'j'} > 0$ by our assumption, so we have $\|n\| \in (1-s, 1)$, so:

$$\frac{\partial P_s(1 - \|n_\star\|)}{\partial n_\star} \neq 0. \tag{22}$$

Combined, we know that:

$$\frac{\partial U_{c\star}^{ij,i'j'}}{\partial n_\star} \neq 0, \tag{23}$$

contradicting the fact that $p_\star$ is optimal.

ii) If $P_s(1 - \|n_\star\|) = 0$, we know by i) that some other terms are positive and the following inequality holds for all positive terms:

$$p_\star^T \frac{\partial P_s(-p_\star^T T_i(d,\theta) x_{ij}^m)}{\partial p_\star} < 0 \tag{24}$$

$$p_\star^T \frac{\partial P_s(p_\star^T T_{i'}(d,\theta) x_{i'j'}^m)}{\partial p_\star} < 0. \tag{25}$$

But since $P_s(1 - \|n_\star\|) = 0$, we have by definition:

$$p_\star^T \frac{\partial P_s(1 - \|n_\star\|)}{\partial p_\star} = 0. \tag{26}$$

Combined, we know that:

$$p_\star^T \frac{\partial U_{c\star}^{ij,i'j'}}{\partial p_\star} < 0, \tag{27}$$

again contradicting the fact that $p_\star$ is optimal. □

Lemma VII.3. *If* $U_c^{ij,i'j'} > 0$*, then i)* $U_{c\star}^{ij,i'j'}$ *is a strictly convex function; and ii)* $U_c^{ij,i'j'}$ *is a locally differentiable function of* d, θ *with:*

$$\frac{dU_c^{ij,i'j'}}{d\big(d,\theta\big)} = \frac{\partial U_{c\star}^{ij,i'j'}(d,\theta,p_\star)}{\partial\big(d,\theta\big)}. \tag{28}$$

Finally, iii) $U_c^{ij,i'j'}$ *is locally twice-differentiable function of* d, θ.

Proof. i) We show that the Hessian of $U_{c\star}^{ij,i'j'}$ at $p_\star$ is non-singular. To this end, we consider a vector $0 \neq q \in \mathbb{R}^4$ and prove that:

$$q^T \frac{\partial^2 U_{c\star}^{ij,i'j'}}{\partial {p_\star}^2} q \neq 0. \tag{29}$$

We establish contradiction otherwise. By Lemma VII.1 ii), we know that each term in $U_{c\star}^{ij,i'j'}$ is convex and contributes a positive semi-definite Hessian matrix. We first derive the Hessian of the term $P_s(1 - \|n_\star\|)$ in the following closed form:

$$\frac{\partial^2 P_s(1 - \|n_\star\|)}{\partial {n_\star}^2} = P_s'' \frac{n_\star}{\|n_\star\|} \frac{n_\star}{\|n_\star\|}^T - P_s' \frac{1}{\|n_\star\|} \left[I - \frac{n_\star}{\|n_\star\|} \frac{n_\star}{\|n_\star\|}^T\right]. \tag{30}$$

By Lemma VII.2 ii), we have:

$$P_s(1 - \|n_\star\|) > 0 \text{ and } \frac{\partial^2 P_s(1 - \|n_\star\|)}{\partial {n_\star}^2} \succ 0. \tag{31}$$

Therefore, the first three elements of q are zero. Since $0 \neq q$, we must have the fourth component of q is non-zero. But by Lemma VII.2 i), we know that one of the other terms in

$U_{c\star}^{ij,i'j'}$ is non-zero at $p_\star$. Without a loss of generality, we can assume:

$$\begin{aligned}&P_s(-p_\star^T T_i(d,\theta)x_{ij}^m) > 0 \text{ and}\\&\frac{\partial^2 P_s(-p_\star^T T_i(d,\theta)x_{ij}^m)}{\partial {p_\star}^2} = P_s''[T_i(d,\theta)x_{ij}^m][T_i(d,\theta)x_{ij}^m]^T \neq 0.\end{aligned} \tag{32}$$

Note we assume homogeneous coordinates for $T_i(d,\theta)x_{ij}^m$. As a result, the fourth component of q must be zero, leading to a contradiction.

ii) Due to i), we can invoke the inverse function theorem, in a local neighborhood, we can assume that $U_c^{ij,i'j'} > 0$, $p_\star$ is a (single-valued) function $p_\star(d,\theta)$, and:

$$\frac{\partial p_\star}{\partial\big(d,\theta\big)} = -\left[\frac{\partial^2 U_{c\star}^{ij,i'j'}}{\partial {p_\star}^2}\right]^{-1}\frac{\partial^2 U_{c\star}^{ij,i'j'}}{\partial p_\star \partial\big(d,\theta\big)}. \tag{33}$$

Note the above derivative is well-defined because our piecewise function P_s is twice-differentiable by construction. By the chain rule, we have:

$$\frac{dU_c^{ij,i'j'}}{d\big(d,\theta\big)} = \frac{\partial U_{c\star}^{ij,i'j'}}{\partial\big(d,\theta\big)} + \frac{\partial U_{c\star}^{ij,i'j'}}{\partial p_\star}\frac{\partial p_\star}{\partial\big(d,\theta\big)} = \frac{\partial U_{c\star}^{ij,i'j'}}{\partial\big(d,\theta\big)}, \tag{34}$$

due to the vanish of gradient at $p_\star$. iii) In the same local neighborhood, we can differentiable the above equation once more to yield:

$$\frac{d^2 U_c^{ij,i'j'}}{d\big(d,\theta\big)^2} = \frac{d}{d\big(d,\theta\big)}\frac{\partial U_{c\star}^{ij,i'j'}}{\partial\big(d,\theta\big)} = \frac{\partial^2 U_{c\star}^{ij,i'j'}}{\partial\big(d,\theta\big)^2} + \frac{\partial^2 U_{c\star}^{ij,i'j'}}{\partial\big(d,\theta\big)\partial p_\star}\frac{\partial p_\star}{\partial\big(d,\theta\big)}. \tag{35}$$

The above step only relies on the second derivatives of $U_{c\star}^{ij,i'j'}$ and is thus well-defined by the construction of P_s. □

B. Global Differentiability

Lemma VII.4. *$U_c^{ij,i'j'}$ is a globally differentiable function of d,θ when $U_c^{ij,i'j'} < \infty$ with the derivative Equation 28 holds everywhere.*

Proof. To extend the property from the local neighborhood, we consider three cases. Case I: If $U_c^{ij,i'j'} > 0$ then differentiability follows from Lemma VII.3. Case II: If $U_c^{ij,i'j'} = 0$ in a local neighborhood, then differentiability follows trivially and agrees with Equation 28 by direct verification. Case III: In the last case, we have $U_c^{ij,i'j'}(d,\theta) = 0$ but there is a sequence of $\big(d_n,\theta_n\big) \to \big(d,\theta\big)$ and $U_c^{ij,i'j'}(d_n,\theta_n) > 0$. In such case, the implicit function theorem fails. To overcome this difficulty, we now define the following auxiliary function:

$$g_n(\beta) = U_c^{ij,i'j'}(\beta(d_n - d) + d, \beta(\theta_n - \theta) + \theta). \tag{36}$$

Clearly, we have $g_n(0) = 0$ and $g_n(1) > 0$. We can then define $\bar\beta = \sup\{\beta | g_n(\beta) = 0\}$. Since $U_c^{iji'j'}$ is a continuous function, we must have $\bar\beta \in (0,1)$. Now since $g_n > 0$ on $(\bar\beta,1)$, g_n' can be computed using Equation 28 on $(\bar\beta,1)$. Applying intermediary value theorem on $(\bar\beta,1)$, we can find $\beta_n \in (\bar\beta,1)$ satisfying:

$$\begin{aligned}&U_c^{ij,i'j'}(d_n,\theta_n) = g_n(1) = g_n'(\beta_n)(1-\bar\beta)\\&= \big((d_n-d)^T,(\theta_n-\theta)^T\big)\frac{dU_c^{ij,i'j'}}{d\big(d,\theta\big)}\Big|_{\big(\beta_n(d_n-d)+d,\beta_n(\theta_n-\theta)+\theta\big)}(1-\bar\beta).\end{aligned} \tag{37}$$

Using this result and we derive the following limit:

$$\begin{aligned}0 \leq &\liminf_{n\to\infty}\frac{U_c^{ij,i'j'}(d_n,\theta_n) - U_c^{ij,i'j'}(d,\theta)}{\|\big(d_n-d,\theta_n-\theta\big)\|}\\ \leq &\limsup_{n\to\infty}\frac{U_c^{ij,i'j'}(d_n,\theta_n) - U_c^{ij,i'j'}(d,\theta)}{\|\big(d_n-d,\theta_n-\theta\big)\|}\\ \leq &\limsup_{n\to\infty}\left\|\frac{dU_c^{ij,i'j'}}{d\big(d,\theta\big)}\Big|_{\big(\beta_n(d_n-d)+d,\beta_n(\theta_n-\theta)+\theta\big)}\right\| = 0,\end{aligned} \tag{38}$$

where the last equality is derived by the fact that the derivative at $\big(\beta_n(d_n-d)+d,\beta_n(\theta_n-\theta)+\theta\big)$ follows from Equation 28, which tends to zero when $U_c^{ij,i'j'}$ approaches zero. Indeed, when $U_c^{ij,i'j'}$ approaches zero, all the terms approach zero consistently, so all the first and second derivatives, i.e. P_s' and P_s'', also tend to zero. With other terms being bounded, we conclude that Equation 28 tends to zero. This result shows that any directional derivative tends to zero in the last case and agrees with Equation 28, thus all is proved. □

C. Global Twice-Differentiability

Next, we prove twice-differentiability using a similar argument as in Lemma VII.4, but we need the following auxiliary result:

Lemma VII.5. *Suppose there is a sequence $\big(d_n,\theta_n\big) \to \big(d,\theta\big)$ such that $U_c^{ij,i'j'}(d_n,\theta_n) > 0$ and $U_c^{ij,i'j'}(d,\theta) = 0$, then:*

$$\lim_{n\to\infty}\frac{d^2 U_c^{ij,i'j'}}{d\big(d,\theta\big)^2}\Big|_{\big(d_n,\theta_n\big)} = 0. \tag{39}$$

Proof. There are two terms in Equation 35 and for the first term:

$$\lim_{n\to\infty}\frac{\partial^2 U_{c\star}^{ij,i'j'}}{\partial\big(d_n,\theta_n\big)^2} = 0, \tag{40}$$

due to the property of P_s. Therefore, our main goal is to show the following second term in Equation 35 also tend to zero, which is largely non-trivial:

$$-\frac{\partial^2 U_{c\star}^{ij,i'j'}}{\partial\big(d_n,\theta_n\big)\partial p_\star}\left[\frac{\partial^2 U_{c\star}^{ij,i'j'}}{\partial {p_\star}^2}\right]^{-1}\frac{\partial^2 U_{c\star}^{ij,i'j'}}{\partial p_\star \partial\big(d_n,\theta_n\big)}. \tag{41}$$

To begin with, we claim the following property, which can be directly verified for our specific $P_s(x) = (x-s)^4/x^5\mathbf{1}(x < s)$:

$$\lim_{x\to s} P_s'/P_s'' = 0 \text{ and } \lim_{x\to s}[P_s'']^{1.5}/P_s' = 6\sqrt{3}/s^{2.5}. \tag{42}$$

Next, we derive the following identity by the optimality of $p_\star$:

$$\begin{aligned}0 =&p_\star^T\frac{\partial U_{c\star}^{ij,i'j'}}{\partial p_\star} = -P_s'(1-\|n_\star\|)\|n_\star\| +\\&\sum_{m=1}^{M}\left[\bar P_s'(-p_\star^T T_i(d_n,\theta_n)x_{ij}^m) + \bar P_s'(p_\star^T T_{i'}(d_n,\theta_n)x_{i'j'}^m)\right].\end{aligned} \tag{43}$$

where we define $\bar P_s'(x) = xP_s'(x)$. Due to Equation 42, as $\lim_{n\to\infty} U_c^{ij,i'j'}(d_n,\theta_n) = 0$, we can choose sufficiently large n such that: $-p_\star^T T_i(d_n,\theta_n)x_{ij}^m > s/2$ for all x_{ij}^m and $p_\star^T T_{i'}(d_n,\theta_n)x_{i'j'}^m > s/2$ for all $x_{i'j'}^m$, $1 > \|n_\star\| > 1 - s$. Finally, because all the terms in Equation 43 are positive, we have the following inequality for x_{ij}^m and $x_{i'j'}^m$ chosen as above:

$$\begin{aligned}&-P_s'(1-\|n_\star\|)\geq -P_s'(1-\|n_\star\|)\|n_\star\|\\ \geq&-[-p_\star^TT_i(d_n,\theta_n)x_{ij}^m]P_s'(-p_\star^TT_i(d_n,\theta_n)x_{ij}^m)\\ \geq&-\frac{s}{2}P_s'(-p_\star^TT_i(d_n,\theta_n)x_{ij}^m)\\ &-P_s'(1-\|n_\star\|)\geq -P_s'(1-\|n_\star\|)\|n_\star\|\\ \geq&-[p_\star^TT_{i'}(d_n,\theta_n)x_{i'j'}^m]P_s'(p_\star^TT_{i'}(d_n,\theta_n)x_{i'j'}^m)\\ \geq&-\frac{s}{2}P_s'(p_\star^TT_{i'}(d_n,\theta_n)x_{i'j'}^m).\end{aligned} \tag{44}$$

Again due to Equation 42, we can choose sufficiently large n such that $P_s''(1-\|n_\star\|)\geq -P_s'(1-\|n_\star\|)$ so that we have the following estimate due to Equation 30:

$$\begin{aligned}\frac{\partial^2P_s(1-\|n_\star\|)}{\partial {n_\star}^2}=&P_s''\frac{n_\star}{\|n_\star\|}\frac{n_\star}{\|n_\star\|}^T-P_s'\frac{1}{\|n_\star\|}\left[I-\frac{n_\star}{\|n_\star\|}\frac{n_\star}{\|n_\star\|}^T\right]\\ \succeq&-P_s'\frac{n_\star}{\|n_\star\|}\frac{n_\star}{\|n_\star\|}^T-P_s'\left[I-\frac{n_\star}{\|n_\star\|}\frac{n_\star}{\|n_\star\|}^T\right]\\ =&P_s'(1-\|n_\star\|)I.\end{aligned} \tag{45}$$

Finally, among all the terms of form $-\bar{P}_s'(-p_\star^TT_i(d_n,\theta_n)x_{ij}^m)$ and $-\bar{P}_s'(p_\star^TT_{i'}(d_n,\theta_n)x_{i'j'}^m)$ in Equation 43, we can assume some term $-\bar{P}_s'(-p_\star^TT_i(d_n,\theta_n)x_{ij}^{m\star})$ achieves the maximal value. (It is also possible that some term $-\bar{P}_s'(p_\star^TT_{i'}(d_n,\theta_n)x_{i'j'}^{m\star})$ on the other convex hull achieves the maximal value, and the case is symmetric.) Once again due to Equation 42 we can choose sufficiently large n such that $P_s''(-p_\star^TT_i(d_n,\theta_n)x_{ij}^{m\star})\geq -P_s'(-p_\star^TT_i(d_n,\theta_n)x_{ij}^{m\star})$. For such maximal term, we have the following estimate:

$$\begin{aligned}&2MsP_s''(-p_\star^TT_i(d_n,\theta_n)x_{ij}^{m\star})\\ \geq&-2MsP_s'(-p_\star^TT_i(d_n,\theta_n)x_{ij}^{m\star})\\ \geq&-2M\bar{P}_s'(-p_\star^TT_i(d_n,\theta_n)x_{ij}^{m\star})\\ \geq&-P_s'(1-\|n_\star\|)\|n_\star\|\geq -P_s'(1-\|n_\star\|)(1-s).\end{aligned} \tag{46}$$

Combining Equation 45 and Equation 46, we have the following estimate of the norm of Hessian:

$$\begin{aligned}\frac{\partial^2U_{c\star}^{ij,i'j'}}{\partial {p_\star}^2}\succeq&\begin{pmatrix}\frac{\partial^2P_s(1-\|n_\star\|)}{\partial {p_\star}^2} & \\ & P_s''(-p_\star^TT_i(d_n,\theta_n)x_{ij}^{m\star})\end{pmatrix}\\ \succeq&-P_s'(1-\|n_\star\|)\min\left[1,\frac{1-s}{2Ms}\right]I,\end{aligned} \tag{47}$$

which implies that:

$$\left\|\left[\frac{\partial^2U_{c\star}^{ij,i'j'}}{\partial {p_\star}^2}\right]^{-1}\right\|\leq\frac{1}{-P_s'(1-\|n_\star\|)\min\left[1,\frac{2Ms}{1-s}\right]}. \tag{48}$$

We are now ready to bound Equation 41 by noting that the mixed derivative can be expanded as:

$$\begin{aligned}&\frac{\partial^2U_{c\star}^{ij,i'j'}}{\partial p_\star\partial\big(d_n,\theta_n\big)}=\\ &\sum_{m=1}^{M}P_s'(-p_\star^TT_i(d_n,\theta_n)x_{ij}^m)\left[-\frac{\partial T_ix_{ij}^m}{\partial\big(d_n,\theta_n\big)}\right]+\\ &\sum_{m=1}^{M}P_s''(-p_\star^TT_i(d_n,\theta_n)x_{ij}^m)\left[-T_ix_{ij}^m\right]\frac{\partial\Big[-p_\star^TT_i(d_n,\theta_n)x_{ij}^m\Big]}{\partial\big(d_n,\theta_n\big)}^T+\\ &\sum_{m=1}^{M}P_s'(p_\star^TT_{i'}(d_n,\theta_n)x_{i'j'}^m)\left[\frac{\partial T_{i'}x_{i'j'}^m}{\partial\big(d_n,\theta_n\big)}\right]+\\ &\sum_{m=1}^{M}P_s''(p_\star^TT_{i'}(d_n,\theta_n)x_{i'j'}^m)\left[T_{i'}x_{i'j'}^m\right]\frac{\partial\Big[p_\star^TT_{i'}(d_n,\theta_n)x_{i'j'}^m\Big]}{\partial\big(d_n,\theta_n\big)}^T.\end{aligned} \tag{49}$$

Combining Equation 48 and with other terms being bounded, we see that Equation 41 is a summation of following terms:

$$\begin{aligned}I\triangleq&O(P_s'(-p_\star^TT_i(d_n,\theta_n)x_{ij}^m)P_s'(-p_\star^TT_{i'}(d_n,\theta_n)x_{i'j'}^{m'})P_s'(1-\|n_\star\|)^{-1})\\ II\triangleq&O(P_s''(-p_\star^TT_i(d_n,\theta_n)x_{ij}^m)P_s'(-p_\star^TT_{i'}(d_n,\theta_n)x_{i'j'}^{m'})P_s'(1-\|n_\star\|)^{-1})\\ III\triangleq&O(P_s''(-p_\star^TT_i(d_n,\theta_n)x_{ij}^m)P_s''(-p_\star^TT_{i'}(d_n,\theta_n)x_{i'j'}^{m'})P_s'(1-\|n_\star\|)^{-1}).\end{aligned} \tag{50}$$

And we show that each term tends to zero, for term of type I, we use Equation 44 to derive:

$$\begin{aligned}I=&O(P_s'(-p_\star^TT_i(d_n,\theta_n)x_{ij}^m)P_s'(-p_\star^TT_{i'}(d_n,\theta_n)x_{i'j'}^{m'})P_s'(1-\|n_\star\|)^{-1})\\ \leq&O(P_s'(-p_\star^TT_i(d_n,\theta_n)x_{ij}^m)P_s'(1-\|n_\star\|)P_s'(1-\|n_\star\|)^{-1})\\ =&O(P_s'(-p_\star^TT_i(d_n,\theta_n)x_{ij}^m))\to 0\end{aligned} \tag{51}$$

For term of type II, we use Equation 44 and choose sufficiently large n to have $P_s''(-p_\star^TT_i(d_n,\theta_n)x_{ij}^m)\geq -P_s'(-p_\star^TT_i(d_n,\theta_n)x_{ij}^m)$ due to Equation 42, yielding the following estimate:

$$\begin{aligned}II=&O(P_s''(-p_\star^TT_i(d_n,\theta_n)x_{ij}^m)P_s'(-p_\star^TT_{i'}(d_n,\theta_n)x_{i'j'}^{m'})P_s'(1-\|n_\star\|)^{-1})\\ \leq&O(P_s'(-p_\star^TT_i(d_n,\theta_n)x_{ij}^m)P_s'(1-\|n_\star\|)P_s'(1-\|n_\star\|)^{-1})\\ =&O(P_s'(-p_\star^TT_i(d_n,\theta_n)x_{ij}^m))\to 0.\end{aligned} \tag{52}$$

For term of type III, Equation 44 and Equation 42 yields:

$$\begin{aligned}III=&O(P_s''(-p_\star^TT_i(d_n,\theta_n)x_{ij}^m)P_s''(-p_\star^TT_{i'}(d_n,\theta_n)x_{i'j'}^{m'})P_s'(1-\|n_\star\|)^{-1})\\ =&O([P_s'(-p_\star^TT_i(d_n,\theta_n)x_{ij}^m)P_s'(-p_\star^TT_{i'}(d_n,\theta_n)x_{i'j'}^{m'})]^{2/3}P_s'(1-\|n_\star\|)^{-1})\\ \leq&O(P_s'(1-\|n_\star\|)^{1/3})\to 0,\end{aligned} \tag{53}$$

thus all is proved. □

Lemma VII.6. *$U_c^{ij,i'j'}$ is a globally twice-differentiable function of d,θ when $U_c^{ij,i'j'}<\infty$ with the second-derivative Equation 35 i) well-defined and ii) holds everywhere.*

Proof. i) Equation 35 is well-defined when $U_c^{ij,i'j'}>0$. When $U_c^{ij,i'j'}=0$, we have:

$$\frac{\partial^2U_{c\star}^{ij,i'j'}}{\partial\big(d,\theta\big)^2}=0\text{ and }\frac{\partial^2U_{c\star}^{ij,i'j'}}{\partial\big(d,\theta\big)\partial p_\star}=0, \tag{54}$$

and Equation 35 is well-defined by the convention $0\cdot\infty=0$.

ii) Following the same argument as Lemma VII.4, we establish twice-differentiability for case I and case II. For case III, where the inverse function theorem fails, define:

$$g_n^k(\beta)=\left[\frac{dU_c^{ij,i'j'}}{d\big(d,\theta\big)}\Big|\Big(\beta(d_n-d)+d,\beta(\theta_n-\theta)+\theta\Big)\right]^k, \tag{55}$$

and invoke the intermediary value theorem to yield:

$$\begin{aligned}g_n^k(\beta)=&\big((d_n-d)^T,(\theta_n-\theta)^T\big)\\ &\left[\frac{d^2U_c^{ij,i'j'}}{d\big(d,\theta\big)^2}\Big|\Big(\beta_n(d_n-d)+d,\beta_n(\theta_n-\theta)+\theta\Big)\right]^k(1-\bar{\beta}),\end{aligned} \tag{56}$$

where $[\bullet]^k$ here indicates the kth element or column of the gradient vector or the Hessian matrix, respectively. The limiting argument leads to:

$$\begin{aligned}0\leq&\liminf_{n\to\infty}\frac{|g_n^k(1)|}{\|\big(d_n-d,\theta_n-\theta\big)\|}\leq\limsup_{n\to\infty}\frac{|g_n^k(1)|}{\|\big(d_n-d,\theta_n-\theta\big)\|}\\ \leq&\limsup_{n\to\infty}\left\|\left[\frac{d^2U_c^{ij,i'j'}}{d\big(d,\theta\big)^2}\Big|\Big(\beta_n(d_n-d)+d,\beta_n(\theta_n-\theta)+\theta\Big)\right]^k\right\|\\ \leq&\limsup_{n\to\infty}\left\|\frac{d^2U_c^{ij,i'j'}}{d\big(d,\theta\big)^2}\Big|\Big(\beta_nd_n,\beta_n\theta_n\Big)\right\|=0,\end{aligned} \tag{57}$$

where the last equality is due to Lemma VII.5, verifying Equation 35 and all is proved. □

D. Momentum Preservation

Finally, by using the globally valid derivative formula, we are ready to show the preservation of linear and angular momentum:

Lemma VII.7. *$U_c^{ij,i'j'}$ preserves i) linear and ii) angular momentum of H_{ij} and $H_{i'j'}$.*

Proof. i) The total negative force applied on H_{ij} and $H_{i'j'}$ is:

$$\begin{aligned}
f_{ij} &\triangleq \sum_{m=1}^{M} \frac{dU_c^{ij,i'j'}}{d[T_i(d,\theta)x_{ij}^m]_3} = \sum_{m=1}^{M} -P_s'(-p_\star^T T_i(d,\theta)x_{ij}^m)n_\star \\
f_{i'j'} &\triangleq \sum_{m=1}^{M} \frac{dU_c^{ij,i'j'}}{d[T_{i'}(d,\theta)x_{i'j'}^m]_3} = \sum_{m=1}^{M} P_s'(p_\star^T T_{i'}(d,\theta)x_{i'j'}^m)n_\star,
\end{aligned}$$

where we have used the derivative Equation 28. Combined, we have:

$$f_{ij} + f_{i'j'} = \frac{\partial U_{c\star}^{ij,i'j'}}{\partial o_\star} n_\star = 0, \tag{58}$$

by the optimality of $p_\star$, proving linear momentum preservation.

ii) The total negative torque applied on H_{ij} and $H_{i'j'}$ is:

$$\begin{aligned}
\tau_{ij} &\triangleq \sum_{m=1}^{M} [T_i(d,\theta)x_{ij}^m]_3 \times \frac{dU_c^{ij,i'j'}}{d[T_i(d,\theta)x_{ij}^m]_3} \\
&= \sum_{m=1}^{M} -P_s'(-p_\star^T T_i(d,\theta)x_{ij}^m)[T_i(d,\theta)x_{ij}^m]_3 \times n_\star \\
\tau_{i'j'} &\triangleq \sum_{m=1}^{M} [T_{i'}(d,\theta)x_{i'j'}^m]_3 \times \frac{dU_c^{ij,i'j'}}{d[T_{i'}(d,\theta)x_{i'j'}^m]_3} \\
&= \sum_{m=1}^{M} P_s'(p_\star^T T_{i'}(d,\theta)x_{i'j'}^m)[T_{i'}(d,\theta)x_{i'j'}^m]_3 \times n_\star,
\end{aligned} \tag{59}$$

where we have used the derivative Equation 28. Combined, we have:

$$\begin{aligned}
\tau_{ij} + \tau_{i'j'} &= \frac{\partial\left[U_{c\star}^{ij,i'j'} - P_s(1-\|n_s\|)\right]}{\partial n_\star} \times n_\star && (60) \\
&= -\frac{\partial P_s(1-\|n_\star\|)}{\partial n_\star} \times n_\star = 0, && (61)
\end{aligned}$$

again by the optimality of $p_\star$, proving angular momentum preservation. □

We summarize our results in the following theorem:

Theorem VII.8. *The contact potential $U_c^{ij,i'j'}$ has the following properties:*

- *$U_c^{ij,i'j'} = 0$ when $dist(H_{ij}, H_{i'j'}) \geq 2s/(1-s)$.*
- *$\lim_{dist(H_{ij},H_{i'j'})\to 0} U_c^{ij,i'j'} = \infty$.*
- *$U_c^{ij,i'j'} = 0$ is globally twice-differentiable in d and θ.*
- *$U_c^{ij,i'j'}$ preserves linear and angular momentum.*

VIII. Well-defined Frictional Damping Potential

Similar to the case with contact potential, we show that the frictional damping potential is also globally twice differentiable and preserves the linear and angular momentum.

Lemma VIII.1. *i) $U_{f\star}^{ij,i'j'}$ is a convex function in $\big(\omega, u\big)$ so $U_f^{ij,i'j'}$ is well-defined; ii) If $U_{f\star}^{ij,i'j'} \not\equiv 0$, $U_{f\star}^{ij,i'j'}$ is strictly convex in u and either strictly convex in ω as well or flat in ω.*

Proof. i) For each term in $U_{f\star}^{ij,i'j'}$, the only function related to $\big(\omega, u\big)$ is $\sqrt{\left\|\dot{x}_{ij,\|}^m - \dot{p}_{ij,\|}^m\right\|^2 + \epsilon}$, which is convex, proving well-definedness.

ii) If any $\mathcal{A}_\epsilon(f_{ij,\perp}^m) > 0$, i.e. $U_{f\star}^{ij,i'j'} \not\equiv 0$, the function has positive definite Hessian in u. If the coefficient of ω, i.e.:

$$[B^{ij,i'j'}]^T[n_\star^t \times [T_i(d,\theta^t)x_{ij}^m]_3] \neq 0,$$

then the Hessian in ω is also positive. If all the coefficients vanish for terms with $\mathcal{A}_\epsilon(f_{ij,\perp}^m) > 0$, then $U_{f\star}^{ij,i'j'}$ is independent of ω, i.e. flat. □

However, although the function is well-defined, the possibility that $U_{f\star}^{ij,i'j'}$ is not strictly convex in ω can significantly complicate our analysis. Fortunately, we can remove such singular situation by shifting the coordinate system along some vector in the tangent plane. For example, let us denote $B_1^{ij,i'j'}$ as the first basis (column) of the tangent projection matrix $B^{ij,i'j'}$. Let us consider the following shifted tangent velocity:

$$\begin{aligned}
\dot{p}_{ij,\|,\alpha}^m(u,\omega) &\triangleq \omega[B^{ij,i'j'}]^T[n_\star^t \times [[T_i(d,\theta^t)x_{ij}^m]_3 + B_1^{ij,i'j'}\alpha]] + u \\
D_{ij,\alpha}^m(d,\theta^{t+1},\theta^t,u,\omega) &\triangleq \mu\Delta t\mathcal{A}_\epsilon(f_{ij,\perp}^m)\sqrt{\left\|\dot{x}_{ij,\|}^m - \dot{p}_{ij,\|,\alpha}^m\right\|^2 + \epsilon} \\
U_{f\star,\alpha}^{ij,i'j'} &\triangleq \sum_m [D_{ij,\alpha}^m + D_{i'j',\alpha}^m] \\
U_{f,\alpha}^{ij,i'j'} &= \min_{u,\omega} U_{f\star,\alpha}^{ij,i'j'}(d,\theta^{t+1},\theta^t,u,\omega),
\end{aligned} \tag{62}$$

where we denote the shifted version by the subscript α. We establish the following result to remove the singular configuration:

Lemma VIII.2. *i) $U_{f,\alpha}^{ij,i'j'} = U_f^{ij,i'j'}$ and, ii) if $U_{f\star,\alpha}^{ij,i'j'} \not\equiv 0$, it is strictly convex for sufficiently large α.*

Proof. i) Let us denote by $u_{\star,\alpha}$ and $\omega_{\star,\alpha}$ the optimal solution in $U_{f,\alpha}^{ij,i'j'}$, then our desired result follows from the following fact:

$$\begin{aligned}
U_{f,\alpha}^{ij,i'j'} \leq& U_{f\star,\alpha}^{ij,i'j'}(d,\theta^{t+1},\theta^t,u_\star - \omega_\star[B^{ij,i'j'}]^T[n_\star^t \times B_1^{ij,i'j'}\alpha],\omega_\star) \\
=& U_{f\star}^{ij,i'j'}(d,\theta^{t+1},\theta^t,u_\star,\omega_\star) = U_f^{ij,i'j'},
\end{aligned} \tag{63}$$

and

$$\begin{aligned}
U_f^{ij,i'j'} \leq& U_{f\star}^{ij,i'j'}(d,\theta^{t+1},\theta^t,u_{\star,\alpha} + \omega_{\star,\alpha}[B^{ij,i'j'}]^T[n_\star^t \times B_1^{ij,i'j'}\alpha],\omega_{\star,\alpha}) \\
=& U_{f\star,\alpha}^{ij,i'j'}(d,\theta^{t+1},\theta^t,u_{\star,\alpha},\omega_{\star,\alpha}) = U_{f,\alpha}^{ij,i'j'}.
\end{aligned} \tag{64}$$

ii) We can set α sufficiently large to make:

$$\|[B^{ij,i'j'}]^T[n_\star^t \times [T_i(d,\theta^t)x_{ij}^m]_3]\| \ll \|[B^{ij,i'j'}]^T[n_\star^t \times [B_1^{ij,i'j'}\alpha]\|, \tag{65}$$

leading to non-zero coefficient of ω in any $\dot{p}_{ij,\|,\alpha}^m$, making $U_{f\star,\alpha}^{ij,i'j'}$ strictly convex in ω if any $\mathcal{A}_\epsilon(f_{ij,\perp}^m) > 0$, i.e. $U_{f\star,\alpha}^{ij,i'j'} \not\equiv 0$. □

A. Local & Global Differentiability

Lemma VIII.3. *If $U_{f\star}^{ij,i'j'}$ is strictly convex in both ω and u, then $U_f^{ij,i'j'}$ is locally differentiable with derivative being:*

$$\frac{dU_f^{ij,i'j'}}{d\big(d,\theta^{t+1},\theta^t\big)}=\frac{\partial U_{f\star}^{ij,i'j'}(d,\theta^{t+1},\theta^t,u_\star,\omega_\star)}{\partial\big(d,\theta^{t+1},\theta^t\big)}. \tag{66}$$

It is also has well-defined mixed-second-derivative being:

$$\begin{aligned}&\frac{d^2U_f^{ij,i'j'}}{d\big(d,\theta^{t+1},\theta^t\big)d\theta^{t+1}}\\=&\frac{\partial^2U_{f\star}^{ij,i'j'}}{\partial\big(d,\theta^{t+1},\theta^t\big)\partial\theta^{t+1}}+\frac{\partial^2U_{f\star}^{ij,i'j'}}{\partial\big(d,\theta^{t+1},\theta^t\big)\partial\big(u_\star,\omega_\star\big)}\frac{\partial\big(u_\star,\omega_\star\big)}{\partial\theta^{t+1}}.\end{aligned} \tag{67}$$

Proof. If $U_{f\star}^{ij,i'j'}$ is strictly convex, then it is strictly convex in a local neighborhood. Invoking the inverse function theorem yields:

$$\begin{aligned}&\frac{\partial\big(u_\star,\omega_\star\big)}{\partial\big(d,\theta^{t+1},\theta^t\big)}=-\left[\frac{\partial^2U_{f\star}^{ij,i'j'}}{\partial\big(u_\star,\omega_\star\big)^2}\right]^{-1}\frac{\partial^2U_{f\star}^{ij,i'j'}}{\partial\big(u_\star,\omega_\star\big)\partial\big(d,\theta^{t+1},\theta^t\big)}\\&\frac{dU_f^{ij,i'j'}}{d\big(d,\theta^{t+1},\theta^t\big)}=\frac{\partial U_{f\star}^{ij,i'j'}}{\partial\big(d,\theta^{t+1},\theta^t\big)}+\frac{\partial U_{f\star}^{ij,i'j'}}{\partial\big(u_\star,\omega_\star\big)}\frac{\partial\big(u_\star,\omega_\star\big)}{\partial\big(d,\theta^{t+1},\theta^t\big)}=\frac{\partial U_{f\star}^{ij,i'j'}}{\partial\big(d,\theta^{t+1},\theta^t\big)},\end{aligned} \tag{68}$$

due to the vanish of gradient at $\big(u_\star,\omega_\star\big)$. The second derivative follows by the chain rule. □

Lemma VIII.4. *If there is a sequence $\big(d_n,\theta_n^{t+1},\theta_n^t\big)\to\big(d,\theta^{t+1},\theta^t\big)$ where $U_{f\star}^{ij,i'j'}(d_n,\theta_n^{t+1},\theta_n^t,u,\omega)\not\equiv 0$ but $U_{f\star}^{ij,i'j'}(d,\theta^{t+1},\theta^t,u,\omega)\equiv 0$ in u and ω, then:*

$$\lim_{n\to\infty}\frac{dU_f^{ij,i'j'}}{d\big(d_n,\theta_n^{t+1},\theta_n^t\big)}=0. \tag{69}$$

Proof. By construction, the only situation where $U_{f\star}^{ij,i'j'}\equiv 0$ is when all $f_{ij,\perp}^m=0$ and all $f_{i'j',\perp}^m=0$. We have the following estimates:

$$\frac{\partial D_{ij}^m}{\partial\theta_n^{t+1}}=\mu\Delta t\mathcal{A}_\epsilon(f_{ij,\perp}^m)\frac{\partial\dot{x}_{ij,\|}^m}{\partial\theta_n^{t+1}}^T\frac{\dot{x}_{ij,\|}^m-\dot{p}_{ij,\|}^m}{\sqrt{\left\|\dot{x}_{ij,\|}^m-\dot{p}_{ij,\|}^m\right\|^2+\epsilon}}\to 0, \tag{70}$$

which is because $\mathcal{A}_\epsilon(f_{ij,\perp}^m)\to 0$ and other terms are bounded. Further, for the derivative with respect to d and θ^t, we have the following estimates:

$$\begin{aligned}\frac{\partial D_{ij}^m}{\partial\big(d_n,\theta_n^t\big)}=&\mu\Delta t\mathcal{A}_\epsilon(f_{ij,\perp}^m)\frac{\partial\dot{x}_{ij,\|}^m}{\partial\big(d_n,\theta_n^t\big)}^T\frac{\dot{x}_{ij,\|}^m-\dot{p}_{ij,\|}^m}{\sqrt{\left\|\dot{x}_{ij,\|}^m-\dot{p}_{ij,\|}^m\right\|^2+\epsilon}}+\\&\mu\Delta t\frac{\partial f_{ij,\perp}^m}{\partial\big(d_n,\theta_n^t\big)}^T\frac{d\mathcal{A}_\epsilon}{df_{ij,\perp}^m}\sqrt{\left\|\dot{x}_{ij,\|}^m-\dot{p}_{ij,\|}^m\right\|^2+\epsilon}\to 0.\end{aligned} \tag{71}$$

Indeed, the first righthand side term above tends to zero because $\mathcal{A}_\epsilon(f_{ij,\perp}^m)\to 0$. The second righthand side term above tends to zero because:

$$\frac{\partial f_{ij,\perp}^m}{\partial\big(d_n,\theta_n^t\big)}=P_s'\frac{\partial n_\star^t}{\partial\big(d_n,\theta_n^t\big)}+P_s''n_\star^t\frac{\partial[p_\star^t]^TT_i(d_n,\theta_n^t)x_{ij}^m}{\partial\big(d_n,\theta_n^t\big)}^T\to 0, \tag{72}$$

with other terms bounded. To verify Equation 72, readers can use the same argument as in Lemma VII.5. By expanding the derivatives in Equation 72 using Equation 33, it can be shown that the derivative is a linear combinations of the following terms:

$$O(P_s'(\bullet)P_s'(\bullet)P_s'(1-\|n_\star\|)^{-1}) \tag{73}$$

$$O(P_s'(\bullet)P_s''(\bullet)P_s'(1-\|n_\star\|)^{-1}) \tag{74}$$

$$O(P_s''(\bullet)P_s''(\bullet)P_s'(1-\|n_\star\|)^{-1}), \tag{75}$$

each of which vanishes. Finally, the same analysis above apply to $D_{i'j'}^m$. Our desired result follows by plugging the above estimates into Equation 66. □

Lemma VIII.5. *$U_f^{ij,i'j'}$ is a globally differentiable function of d,θ^{t+1},θ^t with the derivative Equation 66 holds everywhere.*

Proof. We need to consider several cases. Case I: If $U_{f\star}^{ij,i'j'}$ is strictly convex in both ω and u, then the result follows by Lemma VIII.3. Case II: If $U_{f\star}^{ij,i'j'}$ is strictly convex in u but not ω, then fix a large constant α using Lemma VIII.2 and resort to case I. Case III: If the Hessian is a zero matrix, then $U_{f\star}^{ij,i'j'}\equiv 0$ as a function of ω and u and there are two subcases. Case III.1: If $U_{f\star}^{ij,i'j'}\equiv 0$ in a local neighborhood of $\big(d,\theta^{t+1},\theta^t\big)$, then differentiability follows trivially. Case III.2: If there is a sequence $\big(d_n,\theta_n^{t+1},\theta_n^t\big)\to\big(d,\theta^{t+1},\theta^t\big)$ where $U_{f\star}^{ij,i'j'}(d_n,\theta_n^{t+1},\theta_n^t,u,\omega)\not\equiv 0$, then we can establish differentiability by applying the limiting argument as in Lemma VII.4, yielding:

$$\begin{aligned}0\leq&\liminf_{n\to\infty}\frac{U_f^{ij,i'j'}(d_n,\theta_n^{t+1},\theta_n^t)-U_f^{ij,i'j'}(d,\theta^{t+1},\theta^t)}{\|\big(d_n-d,\theta_n^{t+1}-\theta^{t+1},\theta_n^t-\theta^t\big)\|}\\\leq&\limsup_{n\to\infty}\frac{U_f^{ij,i'j'}(d_n,\theta_n^{t+1},\theta_n^t)-U_f^{ij,i'j'}(d,\theta^{t+1},\theta^t)}{\|\big(d_n-d,\theta_n^{t+1}-\theta^{t+1},\theta_n^t-\theta^t\big)\|}\\\leq&\limsup_{n\to\infty}\left\|\frac{dU_f^{ij,i'j'}}{d\big(d,\theta^{t+1},\theta^t\big)}\Big|_{\big(\beta_n(d_n-d)+d,\beta_n(\theta_n^{t+1}-\theta^{t+1})+\theta^{t+1},\beta_n(\theta_n^t-\theta^t)+\theta^t\big)}\right\|=0,\end{aligned} \tag{76}$$

where the last limit follows from Lemma VIII.4, thus all is proved. □

B. Global Twice-Differentiability

We largely follow Lemma VII.6 to prove twice differentiability, for which we need to establish some limits.

Lemma VIII.6. *If $U_{f\star}^{ij,i'j'}(d,\theta^{t+1},\theta^t,u,\omega)\equiv 0$ in u and ω, then:*

$$\frac{\partial^2U_{f\star}^{ij,i'j'}}{\partial\big(d,\theta^{t+1},\theta^t\big)\partial\big(u_\star,\omega_\star\big)}=0. \tag{77}$$

Proof. We show that each term in the derivative vanish in this case. By taking derivative with respect to $\big(u_\star,\omega_\star\big)$ in Equation 70 and Equation 71, we have:

$$\begin{aligned}&\frac{\partial^2D_{ij}^m}{\partial\theta^{t+1}\partial\big(u_\star,\omega_\star\big)}=O(\mathcal{A}_\epsilon(f_{ij,\perp}^m))=0\\&\frac{\partial^2D_{ij}^m}{\partial\big(d_n,\theta_n^t\big)\partial\big(u_\star,\omega_\star\big)}=O(\mathcal{A}_\epsilon(f_{ij,\perp}^m))+O\left(\left\|\frac{\partial f_{ij,\perp}^m}{\partial\big(d_n,\theta_n^t\big)}\right\|\right)=0,\end{aligned} \tag{78}$$

and all is proved. Here all the terms other than those included in $O(\bullet)$ are bounded and the vanishing of $O(\bullet)$ terms have been proved in Lemma VIII.4. □

Lemma VIII.7. *The mixed-second-derivative Equation 67 i) well-defined and ii) holds everywhere.*

Proof. i) follows from Lemma VIII.6 and the convention that $0\cdot\infty=0$. ii) can be proved by considering the set of cases in Lemma VIII.5.

Case I: If $U_{f\star}^{ij,i'j'}$ is strictly convex in both ω and u, then the result follows by Lemma VIII.3. Case II: If $U_{f\star}^{ij,i'j'}$ is strictly convex in u but not ω, then fix a large constant α using Lemma VIII.2 and resort to case I. Case III: If the Hessian is a zero matrix, then $U_{f\star}^{ij,i'j'} \equiv 0$ as a function of ω and u. Such case can only happen when all the terms of form $\mathcal{A}_\epsilon(f_{ij,\perp}^m) = 0$. But these terms are independent of θ^{t+1}, so $U_{f\star}^{ij,i'j'} \equiv 0$ in a local θ^{t+1}-neighborhood and we immediately have the mixed-second-derivative vanishes due to Lemma VIII.6 and agrees with Equation 67. □

C. Momentum Preservation

We use the globally valid derivative formula to show the preservation of linear and in-plane angular momentum. Unfortunately, we cannot show the full preservation of angular momentum, which is a limitation of our method that is worth further exploration.

Lemma VIII.8. *$U_f^{ij,i'j'}$ preserves i) linear and ii) in-plane angular momentum along $n_\star^t$ of H_{ij} and $H_{i'j'}$.*

Proof. i) The total negative force applied on H_{ij} and $H_{i'j'}$ is:

$$f_{ij} = \sum_{m=1}^{M} \mu\mathcal{A}_\epsilon(f_{ij,\perp}^m) B^{ij,i'j'} \frac{\dot{x}_{ij,\|}^m - \dot{p}_{ij,\|}^m}{\sqrt{\left\|\dot{x}_{ij,\|}^m - \dot{p}_{ij,\|}^m\right\|^2 + \epsilon}} \tag{79}$$

$$f_{i'j'} = \sum_{m=1}^{M} \mu\mathcal{A}_\epsilon(f_{i'j',\perp}^m) B^{ij,i'j'} \frac{\dot{x}_{i'j',\|}^m - \dot{p}_{i'j',\|}^m}{\sqrt{\left\|\dot{x}_{i'j',\|}^m - \dot{p}_{i'j',\|}^m\right\|^2 + \epsilon}}. \tag{80}$$

Combined, we have:

$$f_{ij} + f_{i'j'} = \frac{-1}{\Delta t} B^{ij,i'j'} \frac{\partial U_{f\star}^{ij,i'j'}}{\partial u_\star} = 0, \tag{81}$$

by the optimality of $u_\star$, proving linear momentum preservation.

ii) The total negative torque applied on H_{ij} and $H_{i'j'}$ is:

$$\begin{aligned} \tau_{ij} &= \sum_{m=1}^{M} \mu\mathcal{A}_\epsilon(f_{ij,\perp}^m)[T_i(d,\theta^t)x_{ij}^m] \times B^{ij,i'j'} \frac{\dot{x}_{ij,\|}^m - \dot{p}_{ij,\|}^m}{\sqrt{\left\|\dot{x}_{ij,\|}^m - \dot{p}_{ij,\|}^m\right\|^2 + \epsilon}} \\ \tau_{i'j'} &= \sum_{m=1}^{M} \mu\mathcal{A}_\epsilon(f_{i'j',\perp}^m)[T_{i'}(d,\theta^t)x_{i'j'}^m] \times B^{ij,i'j'} \frac{\dot{x}_{i'j',\|}^m - \dot{p}_{i'j',\|}^m}{\sqrt{\left\|\dot{x}_{i'j',\|}^m - \dot{p}_{i'j',\|}^m\right\|^2 + \epsilon}}. \end{aligned} \tag{82}$$

By the optimality in $\omega_\star$, we have:

$$[n_\star^t]^T(\tau_{ij} + \tau_{i'j'}) = \frac{-1}{\Delta t} \frac{\partial U_{f\star}^{ij,i'j'}}{\partial \omega_\star} = 0, \tag{83}$$

proving the preservation of in-plane angular momentum. □

We summarize our results in the following theorem:

Theorem VIII.9. *The frictional damping potential $U_f^{ij,i'j'}$ has the following properties:*

- *Globally, $U_f^{ij,i'j'} = 0$ is differentiable and has mixed-second-derivative:*

$$\frac{d^2 U_f^{ij,i'j'}}{d\big(d, \theta^{t+1}, \theta^t\big) d\theta^{t+1}}. \tag{84}$$

- *$U_c^{ij,i'j'}$ preserves linear and in-plane angular momentum along $n_\star^t$.*

IX. Differentiability of SDRS

We argument informally in this section. The SDRS function is formulated as an unconstrained optimization, with the objective being:

$$\mathcal{O}(d, \theta^{t+1}, \theta^t, \theta^{t-1}, u^t). \tag{85}$$

At the local minimum, we have:

$$\frac{\partial \mathcal{O}}{\partial \theta^{t+1}} = 0. \tag{86}$$

Now suppose:

$$\frac{\partial^2 \mathcal{O}}{\partial {\theta^{t+1}}^2} \succ 0, \tag{87}$$

and we can invoke the implicit function theorem everywhere, then we have:

$$\frac{\partial \theta^{t+1}}{\partial\big(d, \theta^t, \theta^{t-1}, u^t\big)} = -\left[\frac{\partial^2 \mathcal{O}}{\partial {\theta^{t+1}}^2}\right]^{-1} \frac{\partial^2 \mathcal{O}}{\partial \theta^{t+1} \partial\big(d, \theta^t, \theta^{t-1}, u^t\big)}. \tag{88}$$

Note that all the derivatives in Equation 88 are well-defined due to Theorem VII.8 and Theorem VIII.9. But if we would like to show that θ^{t+1} is globally differentiable, then we need to be able to invoke the implicit function theorem everywhere. A sufficient condition for this is to show that:

$$\frac{\partial^2 \mathcal{O}}{\partial {\theta^{t+1}}^2} \succ 0, \tag{89}$$

everywhere, i.e., $\mathcal{O}$ is a strictly convex function of θ^{t+1} for any choice of $\big(d, \theta^t, \theta^{t-1}, u^t\big)$. We informally argue that this is possible by choosing sufficiently small Δt. Note that in the objective function, only the following term is related to the timestep size:

$$\rho \sum_{m=1}^{M} \frac{\Delta t^2}{2} \|\ddot{x}_{ij}^m\|^2 = O(\Delta t^{-2}), \tag{90}$$

the Hessian matrix of which is called the generalized mass matrix (scaled by Δt^{-2}), which is positive definite. We can choose sufficiently small Δt such that the generalized mass matrix contributes much large positive eigenvalues to $\frac{\partial^2 \mathcal{O}}{\partial {\theta^{t+1}}^2}$ than all other terms, making it positive definite everywhere.

Xiaohan Ye is a Ph.D student at the Computer Graphics and Visualization Lab, The University of Hong Kong. He obtained his B.S. degree from Dalian Maritime University and his M.S. degree from Nankai University. His research focuses primarily on physical simulation and control for robotic systems.

Xifeng Gao is a principal researcher with LIGHTSPEED. He has more than 10 years of academic research experience. He is interested in solving geometric computing related problems in research areas, such as Computer Graphics, Digital Games, CAD/CAE, Multimedia Processing, Robotics, and Digital Fabrication. Please refer to https://gaoxifeng.github.io/ for more details.

Kui Wu is a principal researcher in LIGHTSPEED, Tencent America. Previously, he was a postdoctoral associate in the Computational Design & Fabrication Group under the guidance of Prof. Wojciech Matusik at MIT CSAIL. He received his Ph.D. degree in Computer Science from University of Utah in 2019, advised by Prof. Cem Yuksel. His research interests are computer graphics, especially on mesh processing, knitting, real-time rendering, and physical-based simulation.

Zherong Pan is a principal researcher in LIGHTSPEED. He was a postdoctoral researcher at the Intelligent Motion Lab of the University of Illinois Urbana-Champaign. He obtained Ph.D. degree from the University of North Carolina at Chapel Hill. His research is focused on numerical analysis, operations research, motion planning, and computational physics.

Taku Komura joined Hong Kong University in 2020. Before joining HKU, he worked at the University of Edinburgh (2006-2020), City University of Hong Kong (2002-2006) and RIKEN (2000-2002). He received his BSc, MSc and PhD in Information Science from University of Tokyo. His research has focused on data-driven character animation, physically-based character animation, crowd simulation, 3D modelling, cloth animation, anatomy-based modelling and robotics. Recently, his main research interests have been on physically-based animation and the application of machine learning techniques for animation synthesis. He received the Royal Society Industry Fellowship (2014) and the Google AR/VR Research Award (2017).