Implicit Integration for Articulated Bodies with Contact via the Nonconvex Maximal Dissipation Principle

Zherong Pan and Kris Hauser[†]

***Abstract*— We present non-convex maximal dissipation principle (NMDP), a time integration scheme for articulated bodies with simultaneous contacts. Our scheme resolves contact forces via the maximal dissipation principle (MDP). Whereas prior MDP solvers assume linearized dynamics and integrate using the forward multistep scheme, we consider the coupled system of nonlinear Newton-Euler dynamics and MDP and integrate using the backward integration scheme. We show that the coupled system of equations can be solved efficiently using a novel projected gradient method with guaranteed convergence. We evaluate our method by predicting several locomotion trajectories for a quadruped robot. The results show that our NMDP scheme has several desirable properties including: (1) generalization to novel contact models; (2) stability under large timestep sizes; (3) consistent trajectory generation under varying timestep sizes.**

I. Introduction

Articulated body simulation is an indispensable component of robot motion planning and optimal control. Their governing dynamic equations, i.e. the recursive Newton-Euler's equation [9], and discretization schemes have been studied for decades. However, efficient and accurate contact handling is still a challenging problem despite extensive recent studies [27], [34]. To predict robot motions under simultaneous Coulomb frictional contacts, the two most widely-used formulations are the linear-complementary problem (LCP) [29] and the maximal dissipation principle (MDP) [7]. From a computational perspective, LCP incurs an NP-hard problem while MDP identifies contact forces with the solution of a cheap-to-compute convex program. As reported by [8], MDP-based contact handler achieves the best stability and computational efficiency. Moreover, MDP can encode novel contact models as arbitrary convex wrench spaces, which enables learning contact models from data [33], [34].

The stability region of MDP is empirically shown to be up to ~10 ms according to [31]. Beyond the stability region, the predicted trajectory can either blow-up or drift significantly from the ground truth. Such small stability region not only increases computational cost but also induces problems of vanishing or exploding gradients [16]. In contact-implicit trajectory optimization [21], [28], for example, the problem size grows linearly with the number of timesteps and the cost of a Newton-type method grows superlinearly as a result.

We present a non-convex MDP (NMDP) integrator that: (1) is stable under large timestep sizes; (2) generates consistent contact forces under the MDP formulation; and (3) generalizes to position-dependent contact models. Prior MDP solvers rely on linearized dynamic systems, so that the kinetic energy becomes a quadratic function of the contact forces which can be solved as a convex QP. However, the truncation error of linearization can grow arbitrarily with larger timestep sizes. Our NMDP solver eliminates the truncation error by formulating the *nonlinear* recursive Newton-Euler's equation and the wrench space as a function of the robot pose as a coupled system of nonlinear equations time-integrated using the backward-Euler scheme (Figure 1). The method can inherently account for novel contact models by using the convex shapes as feasible constraints in MDP, with nonlinear dependence on robot pose. To solve this coupled system we propose using the projected gradient method (PGM). We prove that PGM converges under sufficiently small timesteps and show that it empirically converges under large timesteps. We propose an adaptive inner time-integration scheme and guarantee that NMDP solves any (primary) timestep size in finite time.

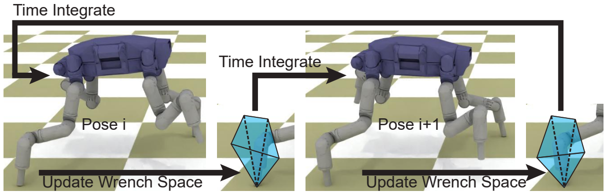

Fig. 1: We consider the three components (robot pose θ update, convex wrench space $v_x(\theta)$ update, and contact force w update) as a coupled system of nonlinear equations, which is solved using a novel projected gradient method with guaranteed convergence.

We evaluate our method by predicting walking and jumping trajectories for the JPL Robosimian and Spider quadruped robot. The results show that NMDP has superior stability under larger timesteps as compared with conventional MDP solvers. In addition, the predicted walking speed and jumping height are more consistent under varying timesteps.

II. Related Work

We review related work on articulated body dynamics, contact handling, and generalized contact models.

Articulated Body Dynamics: Three classes of time discretization schemes have been independently developed for articulated bodies' equation-of-motion. First, variational integrators (VI) [15], [18] discretize the Lagrangian function and then derive the discrete Euler-Lagrange equation. VI preserves momentum and energy symmetry under large timestep sizes. Second, linear multi-step integrators [5] discretize the equivalent Newton-Euler's equation in the configuration space. These integrators are very efficient to evaluate using the Articulated Body Algorithm (ABA) [9]. Third, high-order collocated, position-based integrators use an equivalent

This work is partially funded by NSF Grant #1911087 and authors thank Mengchao Zhang for proofreading the paper. [†] Zherong Pan and Kris Hauser are with the Department of Computer Science, University of Illinois at Urbana-Champaign. {zherong,kkhauser}@illinois.edu

form of the Newton-Euler's equation known as position-based dynamics (PBD) [3], [25], where the main difference is that the discretization is performed in the Euclidean space. Position-based integrators are stable under large timestep sizes but they do not preserve symmetry.

Contact Handling: Sequential contact models [11], [20] have a significantly limited stability region due to the stiffness of contact forces. Models allowing simultaneous collisions and contacts have larger stability region, especially using implicit time stepping [29], [1]. LCP [29], [2] and MDP [7], [14] are the two most popular implicit formulations for simultaneous frictional contacts. Solving complementary conditions due to LCP is NP-hard and can sometimes be infeasible [22]. The MDP relaxes the complementary constraints by allowing any contact forces in the frictional cone to be feasible. However, the stability region of time stepping is still limited by the linearization of dynamic systems using either LCP or MDP. In [1], [25], dynamics with frictionless contacts are reformulated as an optimization and linearization can be avoided, but these results cannot be extended to frictional cases. In [23], the frictional force is modified and then reformulated as an optimization, but the modified variant cannot handle static-sliding frictional mode switches. In [4], a modified two-step Backward Differentiation Formula (BDF2) is proposed to achieve second-order accuracy in time-integration under frictional contacts, but linearization is still needed. Unlike these methods, we analyze the feasibility of contact handling without any linearization for both normal and frictional forces.

Generalized Contact Models: Although the Coulomb frictional model is sufficient for most scenarios involving only rigid objects, other contact models are needed for several reasons. To model the unknown continuous force distribution between a planar object and a flat ground, a general convex wrench space is learned from real-world data in [33]. In other works [34], [17], [32], [13], articulated robots are walking on or swimming in deformable environments with granular or compliant materials. Hu [13] simulated both the granular material and the robot using fine-grained finite element method, which is more than 1000× slower than a standalone articulated body simulation. In [17], [32], the Coulomb frictional model is replaced with analytic and empirical force models. Although these models are cheap to compute, they cannot capture the static-sliding frictional mode switches. Zhu et al. [34] used a similar approach as [33] and learned a robot-pose-dependent contact wrench space. Static-sliding frictional mode switches can thereby be modeled using MDP solver with the learned wrench space as constraints. By extending MDP, our NMDP solver can handle any generalized force models [33], [34] in the form of robot-pose-dependent contact wrench spaces.

III. Articulated Body Dynamics

In this section, we briefly review two prior formulations of articulated body dynamics and their corresponding discretization schemes: the recursive Newton-Euler's equation and position-based dynamics. Both schemes can be extended to derive NMDP solvers.

A. Recursive Newton-Euler's Equation

The continuous Newton-Euler's equation under generalized coordinates takes the following form:

$$0 = H(\theta)\ddot{\theta} + C(\theta,\dot{\theta}) - \sum_{x\in\mathcal{C}} \nabla_\theta X(x,\theta)^T f_x - \tau. \tag{1}$$

Here H is the generalized mass matrix, θ is the robot's configuration vector, $C(\theta,\dot{\theta})$ is the Coriolis and Centrifugal force, $X(x,\theta)$ is the forward kinematic function bringing a point x from the robot's local coordinates to the global coordinates, $\mathcal{C}$ is a set of points in contact with the environment, f_x is the external force on x in world coordinates, and finally τ is the joint torque.

Remark 1: We assume contacts are realized by external forces $f_x \in \mathbb{R}^3$. More general contact models such as [34] require external wrenches $f_x \in \mathbb{R}^6$. In this case, we can replace $\nabla_\theta X(x,\theta)$ with the Jacobian matrix in $\mathbb{R}^{6\times|\theta|}$ and all the following analysis applies accordingly.

To discretize a dynamic system, the linear multistep method uses finite difference approximations for all variables. We illustrate this method with first-order finite difference schemes. Higher-order schemes can be applied following a similar reasoning. We introduce two variables θ_- and θ_{--}. We assume that θ_- is the robot configuration at current time instance, θ_{--} is the robot configuration Δt seconds before (Δt is the timestep size), and θ is the to-be-predicted robot configuration after $\alpha\Delta t$ seconds. Here $\alpha \in (0,1]$ is an additional parameter for timestep size control and we use subscripts to denote functions that are dependent on α. Since NMDP solver requires sufficiently small timestep sizes to converge, we use α to ensure this condition holds. Under these definitions, we can approximate:

$$\dot{\theta} \triangleq \frac{\theta-\theta_-}{\alpha\Delta t} \quad \dot{\theta}_- \triangleq \frac{\theta_- - \theta_{--}}{\Delta t} \quad \ddot{\theta} \triangleq \frac{\dot{\theta}-\dot{\theta}_-}{\Delta t}. \tag{2}$$

Plugging these approximations into the Newton-Euler's equation, the forward-Euler integrator takes the following form:

$$0 = H(\theta_-)\ddot{\theta} - C(\theta_-,\dot{\theta}_-) - \sum_{x\in\mathcal{C}} \nabla_\theta X(x,\theta_-)^T f_x - \tau, \tag{3}$$

which is a linearized dynamic system in θ, f_x, τ. Instead, the backward-Euler integrator evaluates H, C at time-level $\alpha\Delta t$ instead of current time instance, resulting in a nonlinear system of (1) and (2). This system is not guaranteed to have a solution, unless a small enough timestep size is used.

B. Position-Based Dynamics

PBD reformulates the governing equation-of-motion as:

$$0 = \nabla_\theta E_\alpha(\theta, f_x), \tag{4}$$

where we define:

$$E_\alpha(\theta,f) \triangleq I_\alpha(\theta) - \sum_{x\in\mathcal{C}} X(x,\theta)^T f_x - \theta^T\tau$$

$$I_\alpha(\theta) \triangleq \int_{x\in\mathcal{R}} \frac{\rho\|X(x,\theta)-(1+\alpha)X(x,\theta_-)+\alpha X(x,\theta_{--})\|^2}{2\alpha\Delta t^2}dx,$$

and the integral in I_α is taken over the entire robot $\mathcal{R}$. If we assume that θ is a continuous trajectory $\theta(t)$ and $\theta =$

$\theta(t+\alpha\Delta t), \theta_- = \theta(t), \theta_{--} = \theta(t-\Delta t)$, it has been shown in [25] that (4) will converge to (1) as $\Delta t \to 0$. This integral can be evaluated analytically in a similar way as deriving the generalized mass matrix.

Remark 2: I_α is the integral of inertial forces expressed under generalized coordinates. Indeed, by comparing Equation 2 and Equation 1, we immediately see that $\nabla_\theta I_\alpha$ plays the role of $H(\theta_-)\ddot{\theta} - C(\theta_-, \dot{\theta}_-)$. We can prove that these two terms coincide as $\Delta t \to 0$.

Compared with (1)+(2), (4) is always solvable under arbitrarily large timestep sizes because it is integrable. In other words, solving for θ is equivalent to the following optimization:

$$\underset{\theta}{\mathbf{argmin}}\, E_\alpha(\theta, f_x).$$

Note that we assume f_x is a constant in our derivation for the integrability of E_α (i.e. PBD dynamics can be written as $0 = \nabla_\theta E_\alpha(\theta, f_x)$ for some E_α). More generally, PBD can still take an integrable form when the external forces are conservative. In scenarios with dissipative force models such as Coulomb frictional forces, both integrability and PBD's feasibility guarantee are lost, just like Newton-Euler's equation. In this work, we propose an algorithm that solves the system of nonlinearity equations with dissipative force models with guaranteed solvability.

IV. Nonconvex MDP

Our main idea is to combine backward time-integration and frictional contact force computation. In an MDP solver, the force at each contact point f_x belongs to a convex feasible space. We assume that the feasible space is a polytope formed by a set of vertices denoted as v_x^j with $j = 1, \cdots, V_x$. Here V_x is the number of vertices used to model the polytope at contact point x. We assume that all the vertices v_x^j are assembled into a matrix $v_x = \left(v_x^1, \cdots, v_x^{V_x}\right)$. Therefore, every feasible contact force f_x can be represented as:

$$f_x \in \left\{ v_x w_x \middle| w_x \geq 0, \mathbf{1}^T w_x \leq 1 \right\}, \tag{5}$$

where w_x is the weights of convex combination and $\mathbf{1}$ is an all-one vector.

Remark 3: The assumption of feasible contact force being a polytope is essential for our convergence proof. Under this assumption, we will extensively use the property that f_x has a bilinear form of $v_x w_x$, where w_x is bounded and v_x is sufficiently smooth.

A. NMDP Formulation

When modeling an inelastic rigid contact, v_x is set to the vertices of the linearized frictional cone if $X(x,\theta)$ is in contact or penetrating the environment, and v_x is set to zero otherwise. However, the switch between the in-contact and off-contact state is non-differentiable which is undesirable in applications such as differential dynamic programming [31] and trajectory optimization [21]. Therefore, we assume that v_x is a robot-pose-dependent, differentiable function $v_x(\theta)$. This formulation is compatible with the recently proposed learning-based granular wrench space model [34] and can potentially generalize to other contact models. To determine the weights w_x, MDP solves an optimization that minimizes the kinetic energy at time instance $\alpha\Delta t$. Conventional MDP solver uses the linearized dynamic system Equation 3 and discretizes v_x at θ_-, resulting in a QP problem. Instead, our NMDP scheme uses the backward-Euler integrator Equation 1 and discretizes v_x at θ. As a result, we need to solve the following nonlinearly constrained optimization:

$$\begin{gathered}
\underset{\theta,w}{\mathbf{argmin}}\;\; K(\theta) \triangleq \frac{1}{2}\dot{\theta}^T H(\theta)\dot{\theta} \quad \mathbf{s.t.}\; 0 = G_\alpha(\theta, w) \\
G_\alpha(\theta, w) \triangleq H(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) - \sum_{x\in\mathcal{C}} \nabla_\theta X(x,\theta)^T v_x(\theta) w_x - \tau,
\end{gathered} \tag{6}$$

where we assume the use of recursive Newton-Euler's equation and w is a concatenation of all w_x. In the rest of the paper, we propose two algorithms to solve (6).

Remark 4: The solution to Equation 6 might not be unique and a common strategy is to add a regularization of form $\|w\|^2$ and bias the solution towards small contact forces. We do not use this strategy in our implementation, in which case our solvers find an arbitrary solution.

B. NMDP Solver

Since (6) is a general nonlinearly constrained optimization, it can be solved using general-purpose optimizers such as the interior point method [19]. However, these methods are not guaranteed to converge to a first-order stationary point due to infeasibility. Instead, we consider two variants of the projected gradient method (PGM), which we prove to converge to a first order stationary point. PGM starts from a feasible initial guess and updates a search direction of w by solving:

$$\begin{gathered}
\underset{\Delta\theta,\Delta w}{\mathbf{argmin}}\, K(\theta+\Delta\theta) \quad \mathbf{s.t.}\; G_\alpha(\theta+\Delta\theta, w+\Delta w) = 0 \\
K(\theta+\Delta\theta) \triangleq K(\theta) + \nabla_\theta K^T \Delta\theta + \frac{1}{2}\Delta\theta \nabla_\theta^2 K \Delta\theta \\
G_\alpha(\theta+\Delta\theta, w+\Delta w) \triangleq \nabla_\theta G_\alpha \Delta\theta + \nabla_w G \Delta w.
\end{gathered} \tag{7}$$

Since $G_\alpha \equiv 0$, the implicit function theorem implies: $\Delta\theta = -\nabla_\theta {G_\alpha}^{-1} \nabla_w G \Delta w$ where we assume $\nabla_\theta G_\alpha$ is non-singular. We can get an equivalent QP by plugging $\Delta\theta$ into (7):

$$\begin{aligned}
\underset{\Delta w}{\mathbf{argmin}}\; & -\nabla_\theta K^T \nabla_\theta {G_\alpha}^{-1} \nabla_w G \Delta w + \|\Delta w\|^2/\gamma + \\
& \frac{1}{2}\Delta w^T \nabla_w G^T \nabla_\theta {G_\alpha}^{-T} \nabla_\theta^2 K \nabla_\theta {G_\alpha}^{-1} \nabla_w G \Delta w \\
\mathbf{s.t.}\; & (w+\Delta w) \geq 0, \mathbb{1}^T (w+\Delta w) \leq \mathbf{1},
\end{aligned} \tag{8}$$

where we use γ to facilitate line search. The matrix $\mathbb{1}$ is a concatenation of constraints that w_x sums to less than one on each contact point x. After solving for a new $w \leftarrow w + \Delta w$, we update θ by projecting it to the $G_\alpha(\theta, w) = 0$ manifold using the following recursion:

$$\theta \leftarrow \theta - \nabla_\theta {G_\alpha}^{-1} G_\alpha(\theta, w). \tag{9}$$

Note that we have only used the first-order derivatives of G_α in (8) so the PGM has linear convergence speed at best. Our second version of PGM differs in that we ignore all gradients of the function v_x, i.e. zeroth-order update for v_x. This requirement is inspired by the recent work [34] where the contact wrench space is learned from real-world data. In this case, computing derivatives of v_x involves costly backpropagation through a learning model, e.g. neural networks, sublevel sets of high-order polynomials [33], or radial basis functions [34]. Mathematically, the derivatives of v_x only occurs in $\nabla_\theta G_\alpha$ and we denote its zeroth-order, inexact variant as:

$$\nabla_\theta \bar{G}_\alpha(\theta, w) \triangleq \nabla_\theta G_\alpha(\theta, w) + \sum_{x\in\mathcal{C}} \nabla_\theta X(x,\theta)^T \nabla_\theta v_x(\theta) w_x.$$

Using $\nabla_\theta \bar{G}_\alpha$, we derive the following, inexact counterpart of QP (Equation 8):

$$\begin{aligned}\underset{\Delta w}{\mathbf{argmin}}\; & -\nabla_\theta K^T \nabla_\theta \bar{G}_\alpha^{-1} \nabla_w G \Delta w + \|\Delta w\|^2/\gamma + \\ & \frac{1}{2}\Delta w^T \nabla_w G^T \nabla_\theta \bar{G}_\alpha^{-T} \nabla_\theta^2 K \nabla_\theta \bar{G}_\alpha^{-1} \nabla_w \bar{G} \Delta w \\ \mathbf{s.t.}\; & (w+\Delta w) \ge 0, \mathbb{1}^T(w+\Delta w) \le \mathbf{1},\end{aligned} \tag{10}$$

and the following, inexact counterpart of manifold projection (Equation 9):

$$\theta \leftarrow \theta - \nabla_\theta \bar{G}_\alpha^{-1} G_\alpha(\theta, w). \tag{11}$$

The pipeline of both first- and zeroth-order PGM is outlined in Algorithm 1. We compute all the derivatives analytically according to [6] for the Newton-Euler's equation and [25] for the position-based dynamics.

Algorithm 1 (First- / Zeroth-) Order PGM($\alpha, \Delta t, \theta_-, \theta_{--}$)

```
$w^0 \leftarrow 0, \theta^0 \leftarrow \theta_-, \gamma^0 \leftarrow 1, \eta > 1$
while $\|G_\alpha(\theta^0, w^0)\| \neq 0$ do
    Compute (9) or (11) ($\theta = \theta^0, w = w^0$)
for $k = 1, \cdots$ do
    Solve (8) or (10) at ($\theta = \theta^{k-1}, w = w^{k-1}$) for $w^k$
    $\theta^k \leftarrow \theta^{k-1}$
    while $\|G_\alpha(\theta^k, w^k)\| \neq 0$ do
        Compute (9) or (11) at ($\theta = \theta^k, w = w^k$)
    if $K(\theta^k) > K(\theta^{k-1})$ then
        $\gamma \leftarrow \eta\gamma, \theta^k \leftarrow \theta^{k-1}, w^k \leftarrow w^{k-1}$
    else
        $\gamma \leftarrow \gamma/\eta$
        if $\|\theta^k - \theta^{k-1}\|_\infty < \epsilon$ then
            Return $\theta^k, w^k$
```

V. Convergence Analysis

We analyze the convergence of Algorithm 1 in both first- and zeroth-order cases. PGM cannot proceed if $\nabla_\theta G_\alpha$ is rank-deficient and does not have an inverse. In addition, the manifold projection substeps in PGM can diverge without using line-search strategies. Finally, the outer-loop of the zeroth-order PGM can fail to converge by using an inexact gradient. We take the following three assumptions to show that first-order PGM and zeroth-order manifold projection are well-defined and convergent:

***A** 5.1:* $X \in \mathcal{C}^\infty$.

***A** 5.2:* $\sigma_{min}\left[\int_{x\in\mathcal{R}} \frac{\rho}{\Delta t^2} \nabla_\theta X^T \nabla_\theta X dx\right](\theta_-) \ge \sigma_X > 0$.

***A** 5.3:* $\frac{\partial^3 v_x}{\partial \theta^3} \in \mathcal{C}^0$.

To show that zeroth-order PGM is also convergent, we need an additional assumption:

***A** 5.4:* $\nabla_w G$ has full row rank.

Remark 5: A 5.1 and A 5.2 can be satisfied by choosing appropriate parameterizations of robot joints. When all the hinge joints are parameterized using Euler angles, then A 5.1 is satsified. A 5.2 requires that the kinetic energy $K(\theta)$ is strictly convex at θ_-. In other words, for any infinitesimal perturbation $\delta\theta$, there must be some points $x \in \mathcal{R}$ undergoing infinitesimal movements δx with non-vanishing $\lim_{\delta\theta\to 0} \delta x/\delta\theta$. This assumption can only be violated when the robot is suffering from Gimbal lock of Euler angles, which can be easily resolved by moving the singular point away from the current configuration.

Remark 6: A 5.3 indicates that NMDP can only handle smooth contact force models. Stiff and penetration-free contacts between two rigid objects cannot be handled by our method. However, smooth force models are essential for gradient-based motion planning and control [31], [21], [26]. In addition, stiff contacts can be approximated by smooth contacts. For example, a linearized frictional cone is a polytope with vertices being $v_x^i = n + t^i\mu$, where n is the contact normal and t^i is a direction on the tangential plane. We modify this definition to satisfy A 5.3 by setting $v_x^i = d(X(x,\theta))^3(n + t^i\mu)$, where $d(X(x,\theta))$ is the penetration depth of the contact point x. This force model can be made arbitrarily stiff by scaling v_x^i with a big constant.

Remark 7: A 5.4 is a major limitation of the zeroth-order PGM. Note that $v_x = 0$ and thus $\nabla_w G$ has a zero rank when a contact point x does not penetrate the environment. However, once the penetration depth becomes non-zero, then $v_x \neq 0$ and $\nabla_w G$ can have a full row rank. A 5.4 disallows such jumps in the rank of $\nabla_w G$. A simple workaround is to slightly modify the contact model by allowing small, non-zero contact forces even when robot is not in contact with the environment. One method to satisfy A 5.4 is to set $v_x^i = (d(X(x,\theta))^3 + \zeta)(n + t^i\mu)$, where ζ is a small positive constant. In our implementation, we ignore A 5.4 and have never observed divergence behavior due to the violation of this assumption.

Under the above assumptions, our main results are:

Theorem 5.5 (First-Order PGM Convergence): Assuming A 5.1, A 5.2, A 5.3, there exists $\alpha_4 > 0$, such that for all $\alpha \le \alpha_4$, the first order Algorithm 1 will generate a monotonically decreasing sequence of $\{K(\theta^k)\}$ where each θ^k satisfies $G(\theta^k, w^k) = 0$.

Theorem 5.6 (Zeroth-Order PGM Convergence): Assuming A 5.1, A 5.2, A 5.3, A 5.4, there exists $\alpha_5 > 0$, such that for all $\alpha \le \alpha_5$, the zeroth-order Algorithm 1 will generate a monotonically decreasing sequence of $\{K(\theta^k)\}$ where each θ^k satisfies $G(\theta^k, w^k) = 0$.

The proofs of these results are deferred to our extended report [24]. Both results imply that timestep sizes cannot be arbitrarily large, otherwise PGM can fail to converge. In addition, the divergence behavior of PGM can only happen in the manifold projection substep when the norm $\|G_\alpha\|, \|\bar{G}_\alpha\|$ does not decrease after apply Equation 9 or Equation 11, which is an indicator of the use of smaller timestep sizes. As a result, we can design a robust articulated body simulator using adaptive timestep control as illustrated in Algorithm 2.

Algorithm 2 starts by time-integrating using $\alpha = 1$. If PGM diverges, we cut the timestep size by half, i.e. setting $\alpha = 0.5$ and recurse. Note that simulate($\alpha, \Delta t, \theta_-, \theta_{--}$) implies: 1) the last timestep size is Δt, and 2) the desired next time

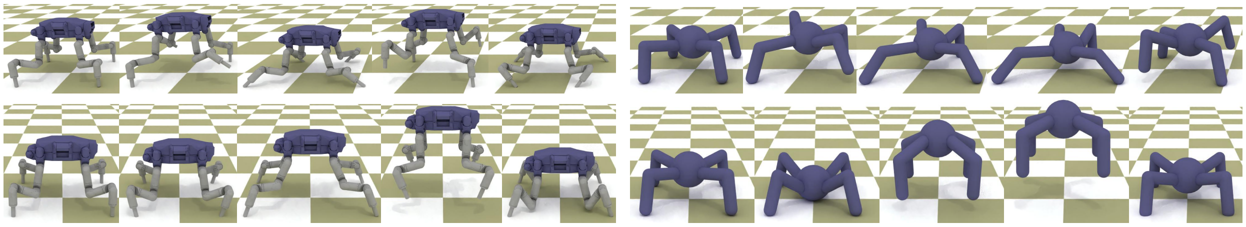

Fig. 2: The robots' walking (top) and jumping (bottom) trajectories tracked using the stable PD controller and simulated using our NMDP solver, where we use $\Delta t = 0.05$ s.

instance is $\alpha\Delta t$ ahead. If PGM diverges, we slice timestep size by half and call simulate$(\alpha/2, \Delta t/2, \theta_-, \theta_{--})$ for the first half. However, further subdivision might happen for the first half due to recursion, so we return the last timestep size, say Δt^*. Next, we time integrate the second half. Given that last timestep size is Δt^* and our desired $\alpha^*\Delta t^* = \alpha\Delta t/2$, we must have $\alpha^* = (\alpha\Delta t)/(2\Delta t^*)$.

Algorithm 2 simulate$(\alpha, \Delta t, \theta_-, \theta_{--})$

```
$\theta, w$,Converged←PGM$(\alpha, \Delta t, \theta_-, \theta_{--})$
**if** Converged **then**
    Return $\alpha\Delta t, \theta, \theta_-$
**else**
    $\Delta t^*, \theta^*, \theta_-^* \leftarrow$ simulate$(\alpha/2, \Delta t, \theta_-, \theta_{--})$
    $\alpha^* \leftarrow (\alpha\Delta t)/(2\Delta t^*)$
    $\Delta t^{**}, \theta^{**}, \theta_-^{**} \leftarrow$ simulate$(\alpha^*, \Delta t^*, \theta^*, \theta_-^*)$
    Return $\Delta t^{**}, \theta^{**}, \theta_-^{**}$
```

A. NMDP Working with PBD

Our analysis and formulation assumes the use of Newton-Euler's equation. An equivalent form of NMDP can be formulated for the position-based dynamics via a new definition of $K(\theta)$ and G_α as follows:

$$
\begin{aligned}
&\mathop{\mathbf{argmin}}_{\theta,w}\; K(\theta) \quad \mathbf{s.t.}\; 0 = G_\alpha(\theta, w) \\
&K(\theta) \triangleq \int_{x\in\mathcal{R}} \frac{\rho\|X(x,\theta) - X(x,\theta_-)\|^2}{2\Delta t^2} dx \qquad (12) \\
&G_\alpha \triangleq \nabla_\theta I_\alpha(\theta) - \sum_{x\in\mathcal{C}} \nabla_\theta X(x,\theta)^T v_x(\theta) w_x - \tau,
\end{aligned}
$$

and all the convergence analysis applies to Equation 6 and Equation 12 alike. We refer readers to our extended report [24] for the proof of Theorem 5.5 and Theorem 5.6.

VI. Evaluations

We evaluate the performance of NMDP variants in challenging locomotion scenarios. We implement NMDP using C++ and Eigen [12], where the optimizations can be solved using both first- and zeroth-order PGM. All the matrix inversions in manifold projection are solved by a rank-revealing LU factorization. As long as the factorization detects that the matrix is near singular (i.e. A 5.2 is violated) or the norm $\|G_\alpha\|$ ($\|\bar{G}_\alpha\|$ in case of ZOPGM) does not decrease, we restart PGM with smaller timestep sizes. In each outer loop of PGM, a QP is solved and the problem data of these QP are quite similar. We use the parametric QP solver qpoases [10] that can make use of these similarities to accelerate computation. We set $\epsilon = 10^{-6}$, $\eta = 1.5$, and $\zeta = 10^{-3}$.

As illustrated in Figure 2, we conduct experiments on the Robosimian by having different simulators to track a prescribed robot walking or jumping trajectory using the stable PD controller described in Ref. [30]. The stable PD controller is consistent with the backward-Euler integrator, which uses $\theta, \dot{\theta}$ instead of $\theta_-, \dot{\theta}_-$ as the target state to be tracked. We compare the performance of the following simulators:

- NE-NMDP/NE-ZONMDP: Newton-Euler NMDP solved using first-/zeroth-order PGM.
- PBD-NMDP/PBD-ZONMDP: Position-based dynamics NMDP solved using first-/zeroth-order PGM.
- NE-MDP: Baseline method, performs forward integration of Newton-Euler's equation with contact forces solved using MDP.

Stability Under Large Timestep Sizes: We track a robot jumping trajectory that uses symmetric poses. Since the Robosimian's body shape is also symmetric, the torso in the tracked trajectory should have zero tilt angles from the vertical axis, which is our groundtruth. In Figure 3 (a, b), we plot the torso's tilt angle predicted by each simulator under different timestep sizes. NE-MDP is only stable when $\Delta t \le 7$ ms and the simulator explodes under larger Δt. Even when $\Delta t \le 7$ ms, the tilt angle suffers from severe oscillations. In comparison, NE-NMDP is stable when Δt increases from 5 ms to 50 ms and the predicted tilt angle oscillation is relatively small. We further plot the tilt angle sequence predicted by the other three variants of NMDP in Figure 3 (c, d, e), the oscillations are consistently small. First-order NMDP with $\Delta t \le 50$ ms, never experienced any divergence behaviors so Algorithm 2 is never needed. But divergence happens in the zero-order case and we observe larger tilt angle oscillation in Figure 3 (c, e).

Consistent Prediction: The tracked robot jumping trajectory lasts for 10 seconds with 5 repeated jumping behaviors. As a result, the expected torso height should be a periodic function. In Figure 4, we plot the torso height trajectory predicted using the five methods. With $\Delta t \le 7$ ms, the trajectories generated by NE-MDP are suffering from relatively large variations. While all four NMDP variants can consistently predict a periodic function when Δt increases from 5 ms to 50 ms. A similar result is observed in the walking trajectory. As shown in Figure 5, the groundtruth walking distance is a linear function of time. The trajectories predicted using NE-MDP exhibit a large variations with different timestep sizes. The consistency of NE-(ZO)NMDP are much better and that of PBD-(ZO)NMDP are the best. Compared with NE-(ZO)NMDP, the additional stability and consistency of PBD-(ZO)NMDP are presumably due to the fact that Euclidean space discretization is more accurate

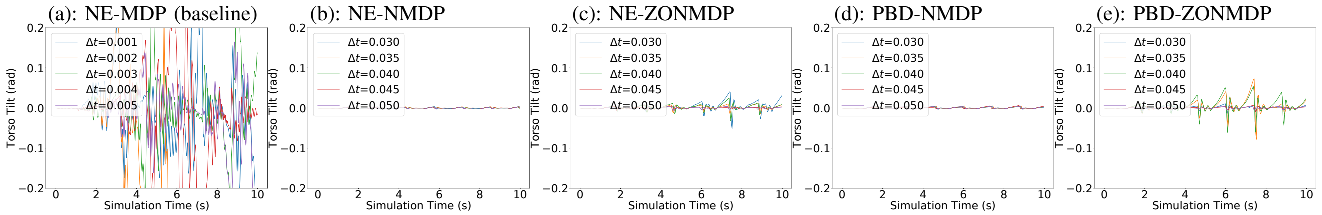

Fig. 3: The torso tilt of the tracked jumping trajectory.

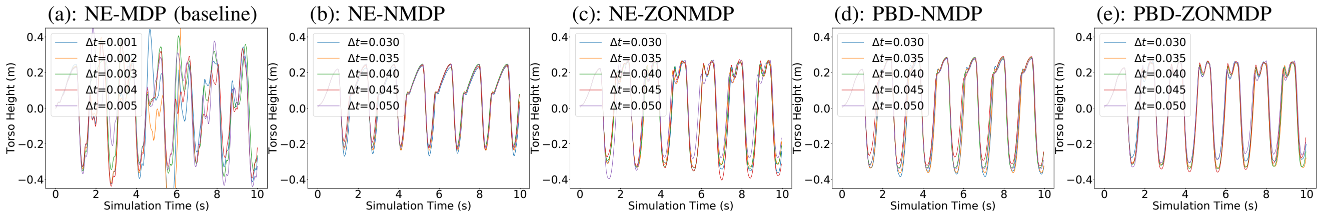

Fig. 4: The torso height of the tracked jumping trajectory.

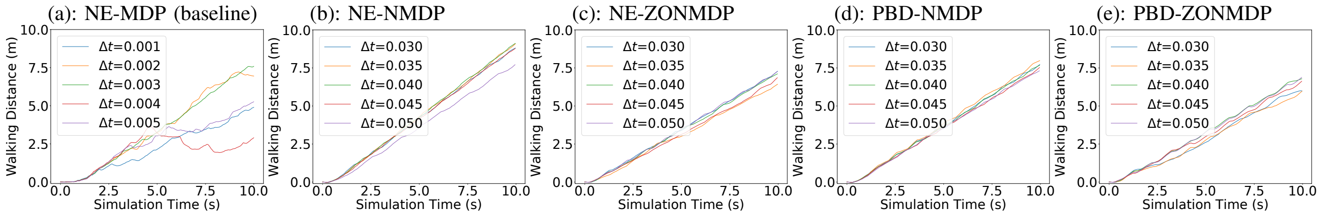

Fig. 5: The distance of the tracked walking trajectory.

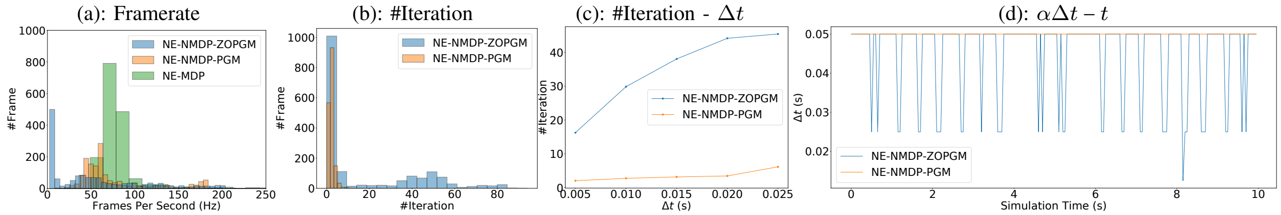

Fig. 6: A performance comparison of different methods. (a): Histogram of instantaneous framerates (1/(wall clock time per frame)), (b): Histogram of number of outer iterations per frame, (c): Number of outer iterations against timestep size (Δt), (d): Instantaneous adaptive timestep size ($\alpha\Delta t$) chosen by Algorithm 2 plotted against simulation time (t).

TABLE I: Average std. dev. of trajectories between different Δt

Quantity	NE-MDP	PBD-NMDP	NE-NMDP
Robosmian-Walk-Dist	0.72	0.25	**0.10**
Robosmian-Jump-Height	0.26	**0.01**	0.02
Spider-Walk-Dist	**0.09**	**0.09**	0.10
Spider-Jump-Height	0.13	**0.04**	0.05

than that in the configuration space [25]. We summarize the standard deviation of trajectories simulated using different Δt in Table I.

Computational Cost: We summarize the computational performance by collecting and analyzing all the timesteps in the four trajectories of Figure 2 simulated at $\Delta t = 50$ ms. Figure 6 (a) profiles the instantaneous framerate, of which MDP is the most efficient involving a single QP solve. FOPGM does not incur a significant sacrifice in framerate while ZOPGM is significantly slower. Figure 6 (b) profiles the number of outer iterations of Algorithm 1 until convergence and Figure 6 (c) plots the average number of outer iterations against Δt. These figures show that ZOPGM is approximately one order of magnitude slower than FOPGM, but ZOPGM provides the extra convenience that analytic derivatives of $v_x(\theta)$ are not needed. Finally, Figure 6 (d) shows the smallest timestep size chosen by Algorithm 2 for FOPGM and ZOPGM. At $\Delta t = 50$ ms, PGM is always convergent. ZOPGM is also convergent, but $\alpha\Delta t$ needs to get down to 25 ms during some critical time instances (e.g. when robot changes contact states).

VII. Conclusion & Discussion

We present a backward-Euler time integration scheme for articulated bodies under generalized contact models. Our key idea is the representation of contact forces as a convex combination of vertices of the feasible contact wrench space. A backward-Euler integration scheme discretizes both the articulated body dynamics and the vertices of contact wrench spaces at the next time instance instead of the current one.

Our method has several limitations. First, we assume a set of known contact points, and our current implementation does not handle the case where contacts are introduced or broken within a timestep. Second, our generalized contact model needs to be differentiable, for which we allow robots to penetrate the ground. Our method does not apply to contact models that prevent collisions exactly. Third, a fundamental assumption of our convergence analysis is that w is bounded, so we need to know the maximal contact force and our method cannot handle infinite Coulomb's frictional cones. Finally, our convergence analysis assumes that $G_\alpha = 0$ which is impossible in practice. In the future, we plan to analyze the convergence of our method under finite precision.

References

[1] M. Anitescu, "A fixed time-step approach for multibody dynamics with contact and friction," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 4, 2003, pp. 3725–3731 vol.3.

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

[3] J. Bender, M. Müller, and M. Macklin, "A Survey on Position Based Dynamics," in *EG 2017 - Tutorials*, A. Bousseau and D. Gutierrez, Eds. The Eurographics Association, 2017.

[4] G. E. Brown, M. Overby, Z. Forootaninia, and R. Narain, "Accurate dissipative forces in optimization integrators," *ACM Trans. Graph.*, vol. 37, no. 6, Dec. 2018. [Online]. Available: https://doi.org/10.1145/3272127.3275011

[5] J. C. Butcher and N. Goodwin, *Numerical methods for ordinary differential equations*. Wiley Online Library, 2008, vol. 2.

[6] J. Carpentier and N. Mansard, "Analytical Derivatives of Rigid Body Dynamics Algorithms," in *Robotics: Science and Systems (RSS 2018)*, Pittsburgh, United States, June 2018. [Online]. Available: https://hal.laas.fr/hal-01790971

[7] E. Drumwright and D. A. Shell, "Modeling contact friction and joint friction in dynamic robotic simulation using the principle of maximum dissipation," in *Algorithmic foundations of robotics IX*. Springer, 2010, pp. 249–266.

[8] 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.

[9] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

[10] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpoases: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[11] 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.

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

[13] Y. Hu, Y. Fang, Z. Ge, Z. Qu, Y. Zhu, A. Pradhana, and C. Jiang, "A moving least squares material point method with displacement discontinuity and two-way rigid body coupling," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.

[14] 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.

[15] J. Lee, C. K. Liu, F. C. Park, and S. S. Srinivasa, "A linear-time variational integrator for multibody systems," in *Algorithmic Foundations of Robotics XII*. Springer, 2020, pp. 352–367.

[16] G.-H. Liu, T. Chen, and E. A. Theodorou, "Differential dynamic programming neural optimizer," *arXiv preprint arXiv:2002.08809*, 2020.

[17] R. D. Maladen, Y. Ding, P. B. Umbanhowar, A. Kamor, and D. I. Goldman, "Biophysically inspired development of a sand-swimming robot," in *Robotics: Science and Systems*. Georgia Institute of Technology, 2011.

[18] J. Marsden and M. West, "Discrete mechanics and variational integrators," *Act Numerica*, vol. 10, no. 5, pp. 357–514, 2001.

[19] S. Mehrotra, "On the implementation of a primal-dual interior point method," *SIAM Journal on optimization*, vol. 2, no. 4, pp. 575–601, 1992.

[20] B. V. Mirtich, *Impulse-based dynamic simulation of rigid body systems*. University of California, Berkeley, 1996.

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

[22] P. Painlevé, "Sur les lois du frottement de glissement," *Nonlinear Dynamics*, vol. 8, no. 5, pp. 977–979, 2012.

[23] Z. Pan and D. Manocha, "Position-based time-integrator for frictional articulated body dynamics," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–8.

[24] Z. Pan and K. Hauser, "Implicit integration for articulated bodies with contact via the nonconvex maximal dissipation principle," *arXiv:2010.14691*, p. https://arxiv.org/abs/2010.14691, 2020.

[25] Z. Pan and D. Manocha, "Time Integrating Articulated Body Dynamics Using Position-Based Collocation Methods," in *Algorithmic Foundations of Robotics XIII*, M. Morales, L. Tapia, G. Sánchez-Ante, and S. Hutchinson, Eds. Cham: Springer International Publishing, 2020, pp. 673–688.

[26] 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. [Online]. Available: https://doi.org/10.1145/3309486.3340246

[27] T. Preclik, S. Eibl, and U. Rüde, "The maximum dissipation principle in rigid-body dynamics with inelastic impacts," *Computational Mechanics*, vol. 62, no. 1, pp. 81–96, 2018.

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

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

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

[31] 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.

[32] V. Vasilopoulos, I. S. Paraskevas, and E. G. Papadopoulos, "Compliant terrain legged locomotion using a viscoplastic approach," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4849–4854.

[33] J. Zhou, M. T. Mason, R. Paolini, and D. Bagnell, "A convex polynomial model for planar sliding mechanics: theory, application, and experimental validation," *International Journal of Robotics Research*, 2018.

[34] Y. Zhu, L. Abdulmajeid, and K. Hauser, "A data-driven approach for fast simulation of robot locomotion on granular media," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7653–7659.