

GPU-Accelerated Continuous-Time Successive Convexification for Contact-Implicit Legged Locomotion

Samuel C. Buckner¹ and Purnanand Elango^{2*}

Abstract—Contact-implicit trajectory optimization (CITO) enables the automatic discovery of contact sequences, but most methods rely on fine time discretization to capture all contact events accurately, which increases problem size and runtime while tying solution quality to grid resolution. We extend the recently proposed sequential convex programming (SCP) approach for trajectory optimization, continuous-time successive convexification (CT-SCvx), to CITO by introducing integral cross-complementarity constraints, which eliminate the risk of missing contact events between discretization nodes while preserving the flexibility of contact mode changes. The resulting framework, contact-implicit successive convexification (CI-SCvx), models full multibody dynamics in maximal coordinates, including stick-slip friction and partially elastic impacts. To handle complementarity constraints, we embed a backtracking homotopy scheme within SCP for reliable convergence. We implement this framework in a stand-alone Python software, leveraging JAX for GPU acceleration and a custom canonical-form parser for the convex subproblems of SCP to avoid the overhead of general-purpose modeling tools such as CVXPY. We demonstrate CI-SCvx on diverse legged-locomotion tasks. In particular, we validate the approach in MuJoCo with the Gymnasium HalfCheetah model against the MuJoCo MPC baseline, showing that a tracking simulation with the optimized torque profiles from CI-SCvx produces physically consistent trajectories with lesser energy consumption. We also show that the resulting software achieves faster solve times than existing state-of-the-art SCP implementations by over an order of magnitude, thereby demonstrating a practically important contribution to scalable real-time trajectory optimization.

I. INTRODUCTION

Trajectory optimization is a central tool for generating agile behaviors in legged robots [1]. Contact-implicit trajectory optimization (CITO) methods are particularly attractive because they eliminate the need to predefine contact schedules. However, most existing CITO formulations rely on fine time discretization to capture contact events. This increases the number of decision variables and constraints, coupling solution quality tightly to the discretization resolution and creating an unfavorable accuracy-runtime tradeoff [2].

To address this limitation, we build on the sequential convex programming (SCP) approach, which iteratively approximates and solves nonconvex problems. While the usage of SCP has been addressed for CITO [3, 4], we instead consider the recently proposed continuous-time successive convexification (CT-SCvx) framework [5], which notably enforces constraints in continuous time and thus permits

high-fidelity, feasible solutions over coarse discretization grids. Our key idea is to extend continuous-time satisfaction to contact constraints, specifically cross-complementarity constraints [6]. We introduce an integral form of cross-complementarity constraints expressed in continuous time, which ensures that contact mode changes occur at discretization nodes while eliminating the risk of missing contact events between nodes. In conjunction with time-dilation [5], where nodes are allowed to stretch or shrink over time, the generality of the resulting contact mode sequence is ensured.

We formulate the full multibody dynamics of a robot in maximal coordinates rather than using reduced-order approximations such as SRBD [1] or centroidal dynamics [7]. Although maximal coordinates enlarge the state dimension, the associated dynamics and constraints are highly sparse, enabling efficient GPU parallelization. The model supports stick-slip friction, consistent with the maximum dissipation principle, and partially elastic impacts [8]. To handle complementarity constraints, we employ a homotopy scheme with backtracking that tightens the relaxation during the SCP iterations, ensuring reliable convergence. We implement the approach in stand-alone Python software that leverages JAX for GPU-accelerated differentiation and linearization, together with a custom parser for the convex subproblems in SCP that avoids the overhead of general-purpose modeling tools such as CVXPY. We demonstrate the approach on two-dimensional legged robot models, including the Gymnasium HalfCheetah model [9].

A. Related Work

CITO has been studied extensively to generate locomotion solutions without prescribing contact schedules. Early approaches used direct collocation with complementarity constraints to enforce non-penetration and frictional contact [10], while contact-invariant optimization [11] handled contact through smooth penalties that encourage sticking and end-effector alignment, with dynamics enforced softly. These methods typically require fine time discretization to capture contact events accurately, yielding large-scale problems whose solution quality depends strongly on the grid resolution. To mitigate the risk of missing contact events between nodes, cross-complementarity constraints were proposed [6]. More recent formulations such as FESD-J [12] and moving finite elements [13] extend this idea by enabling exact switch detection for systems with impacts and Coulomb friction, ensuring that contact events occur at finite-element boundaries. However, these are still inherently discrete-time constructions.

¹Department of Aeronautics and Astronautics, University of Washington, Seattle, WA 98195.

S. C. Buckner interned at MERL during the development of this work.

²Mitsubishi Electric Research Laboratories, Cambridge, MA 02139.

*Corresponding author. Email: elango@merl.com

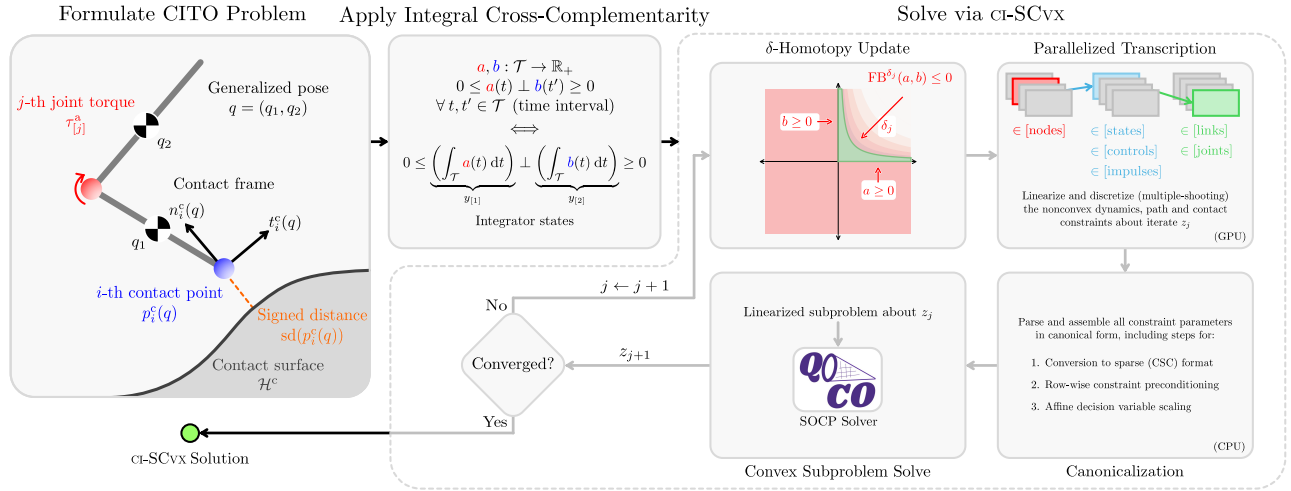


Fig. 1: Overview of the proposed framework, including problem formulation, integral cross-complementarity and the ci-SCvx algorithm.

Continuous-time trajectory optimization frameworks like *ct-SCvx* [5] enforce constraints in continuous time and thereby achieve accurate solutions with coarse discretization, but they have not been extended to CITO. In terms of modeling, many works adopt reduced-order approximations such as centroidal dynamics [7] or single-rigid-body dynamics (SRBD) [1]. While these simplifications can reduce the size of optimization problems and sometimes eliminate the risk of missing contact events [14], they often neglect whole-body effects and can lead to conservative solutions that underutilize the system’s full range of motion.

B. Contributions

This work introduces a continuous-time CITO framework for locomotion tasks and makes the following contributions:

- 1) The contact-implicit successive convexification (ci-SCvx) framework, an SCP-based approach for CITO that ensures continuous-time constraint satisfaction.
- 2) Integral cross-complementarity constraints that ensure contact mode changes occur at discretization nodes. Combined with time-dilation [5], this preserves generality in contact mode sequences.
- 3) A backtracking homotopy scheme that progressively tightens the complementarity constraint relaxation within SCP iterations, ensuring reliable convergence.
- 4) A stand-alone Python software that leverages JAX for GPU-accelerated differentiation and linearization, together with a custom parser for the convex subproblems in SCP that avoids the overhead of general-purpose modeling tools such as CVXPY.
- 5) Demonstration on two-dimensional locomotion tasks, including the Gymnasium HalfCheetah model [9], with comparisons to a state-of-the-art baseline.

II. DYNAMICAL MODEL WITH CONTACT AND IMPACT

We adopt the following notation for the remainder of the development. Concatenation of vectors $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^m$ is $(u, v) \in \mathbb{R}^{n+m}$. The scalar components of $v \in \mathbb{R}^n$ are denoted by $v_{[i]}$, for $i = 1, \dots, n$. The ramp

(ReLU) function is denoted by $|x|_+$. Smooth pseudo-Huber approximations of the Euclidean norm, absolute value, and ramp functions with unit steepness parameter are denoted by $\|x\|_\sim$, $|x|_\sim$, and $|x|_\nabla$. For vector arguments, $|x|_\nabla$ is understood componentwise.

We use maximal coordinates to model the dynamics of a multi-link robot. Let $q \in \mathbb{R}^{n^q}$ denote the generalized position (pose) and $v \in \mathbb{R}^{n^q}$ the generalized velocity. The equations of motion are obtained by applying Newton’s second law to each link, accounting for forces and torques from gravity, joint reaction forces, joint torques, joint stiffness and damping, and contact forces. The robot has n^c contact points with positions, expressed in the inertial (world) frame, given by $p_i^c(q) \in \mathbb{R}^{n^d}$ for $i = 1, \dots, n^c$. The Jacobian of the i th contact point (the contact Jacobian) is $J_i^c(q) = \nabla p_i^c(q)$.

Contact occurs on a smooth surface $\mathcal{H}^c = \{p \in \mathbb{R}^{n^d} \mid h^c(p) = 0\}$, where h^c is a continuously differentiable function and $n^d \in \{2, 3\}$ is the dimension of the position space. At each contact point, we define a contact frame consisting of the unit surface normal $n_i^c(q) = \nabla \text{sd}(p_i^c(q))^\top \in \mathbb{R}^{n^d}$ and $(n^d - 1)$ unit tangent vectors given by the columns of $t_i^c(q) \in \mathbb{R}^{n^d \times (n^d - 1)}$. Together, the columns of $t_i^c(q)$ and $n_i^c(q)$ form a right-handed basis. The rotation matrix that maps vectors from the contact frame to the world frame is $R_i^c(q) = [t_i^c(q) \ n_i^c(q)]$. Fig. 1 shows the frame definitions. The contact force at the i th contact point, represented in the contact frame, is $\phi_i^c = (\phi_i^{c,t}, \phi_i^{c,n})$, where $\phi_i^{c,n} \in \mathbb{R}$ is the normal component along $n_i^c(q)$ and $\phi_i^{c,t} \in \mathbb{R}^{n^d - 1}$ is the tangential component. Similarly, the contact impulse due to impact is $\Phi_i^c = (\Phi_i^{c,t}, \Phi_i^{c,n})$ with the same component-wise interpretation.

Let $c^j(q) = 0_{n_j}$ impose the kinematic constraints that connect links at the joints and restrict degrees of freedom according to joint type. The effect of joint reaction forces on the generalized acceleration \dot{v} is captured by $\nabla c^j(q)^\top \phi^j$, where ϕ^j concatenates all joint reaction forces in the world frame. An impact at any contact point induces a joint reaction impulse that instantaneously changes the generalized

velocity via $\nabla c^j(q)^\top \Phi^j$, where Φ^j concatenates joint reaction impulses in the world frame.

A. Continuous-Time Model

The equations of motion are given by

$$\begin{aligned} \dot{q} &= v, & (1a) \\ M(q)\dot{v} &= W(q, v, \tau^a, a^g) + J^c(q)^\top R_i^c(q)\phi^c \\ &\quad + \nabla c^j(q)^\top \Phi^j, & (1b) \end{aligned}$$

where $J^c(q)^\top = [J_1^c(q)^\top \dots J_{n^c}^c(q)^\top]$, $R^c(q) = \text{blockdiag}(R_1^c(q), \dots, R_{n^c}^c(q))$, and $\phi^c = (\phi_1^c, \dots, \phi_{n^c}^c)$. $M(q)$ is the mass matrix, and $W(q, v, \tau^a, a^g)$ collects the generalized forces and torques due to gravity (with a^g the gravitational acceleration), joint torques τ^a , and joint stiffness and damping. To model physically realistic contact, we require (i) a no-penetration condition expressed as a complementarity constraint between the normal contact force and signed distance to the contact surface, and (ii) a tangential contact force determined by the maximum dissipation principle subject to a friction cone constraint

$$0 \leq \phi^{c,n} \perp \text{sd}(p_i^c(q)) \geq 0, \quad (2a)$$

$$\phi^{c,t} \in \underset{\phi^t}{\text{argmin}} \phi^{c,t \top} v_i^{c,t}(q, v) \quad \text{s.t.} \quad \|\phi^t\| \leq \mu_i \phi_i^{c,n}, \quad (2b)$$

for $i = 1, \dots, n^c$, where μ_i is the friction coefficient and $v_i^{c,t}(q, v) = t_i^c(q)^\top J_i^c(q)v$ is the tangential velocity at the i th contact. The first-order optimality conditions for (2b) are

$$0_{n^d-1} = v_i^{c,t}(q, v) + 2\gamma_i \phi_i^{c,t}, \quad (3a)$$

$$0 \leq \gamma_i \perp (\mu_i \phi_i^{c,n})^2 - |\phi_i^{c,t}|^2 \geq 0, \quad (3b)$$

where γ_i is the Lagrange multiplier and we denote $\gamma = (\gamma_1, \dots, \gamma_{n^c})$. The friction cone constraint is written in squared form to ensure differentiability at the boundary. When the i th contact is active (i.e., signed distance is zero), γ_i is a continuous function of time. Two issues arise with (3): (i) the $\|\cdot\|^2$ term causes numerical ill-conditioning, and (ii) when the i th contact is inactive (i.e., signed distance is positive), $\phi_i^c = 0_{n^d}$, so (3a) reduces to the non-physical case that the contact point's tangential velocity is zero, i.e., $t_i^c(q)^\top J_i^c(q)v = 0_{n^d-1}$. We address both by rewriting (3) as

$$0_{n^d-1} = \phi_i^{c,n} (v_i^{c,t}(q, v) + \gamma_i \nabla \|\phi_i^{c,t}\|^\top), \quad (4a)$$

$$0 \leq \gamma_i \perp |\mu_i \phi_i^{c,n}|_\sim - \|\phi_i^{c,t}\|_\sim \geq 0, \quad (4b)$$

using the smooth norm $\|\cdot\|_\sim$ and smooth absolute value $|\cdot|_\sim$. The multiplication of $\phi_i^{c,n}$ in (4a) enforces stationarity only when contact is active, i.e., normal contact force is positive. Henceforth, we denote $\lambda_i^c(q, v, \phi_i^c, \gamma_i) = v_i^{c,t}(q, v) + \gamma_i \nabla \|\phi_i^{c,t}\|_\sim^\top$ and $\rho_i^c(\phi_i^c, \gamma_i) = |\mu_i \phi_i^{c,n}|_\sim - \|\phi_i^{c,t}\|_\sim$.

Next, we eliminate the explicit joint reaction force in (1b) by enforcing joint constraints at the acceleration level

$$c^j(q) = 0_{n^j} \implies \nabla c^j(q)v = 0_{n^j}, \quad (5a)$$

$$\implies \nabla c^j(q)\dot{v} + (\dots v^\top \nabla^2 c_{[i]}^j(q)v \dots) = 0_{n^j}, \quad (5b)$$

where $c_{[i]}^j(q)$ is the i th component of c^j , for $i = 1, \dots, n^j$. From (5b) and (1b) we solve for ϕ^j and substitute into (1b).

We rewrite (1) in state-space form as $\dot{x} = f(x, u)$, with state $x = (q, v) \in \mathbb{R}^{n^x}$, control input $u = (\tau^a, \phi^c, \gamma) \in \mathbb{R}^{n^u}$, and dimensions $n^x = 2n^q$ and $n^u = n^a + n^c(n^d + 1)$.

The state and control input are subject to path constraints $g(x(t), u(t)) \leq 0_{n^g}$ for all $t \in [0, t^f]$, which include, for each $i = 1, \dots, n^c$, nonnegativity constraints on $\phi_i^{c,n}$, $\text{sd}(p_i^c(q))$, γ_i , and $\rho_i^c(\phi_i^c, \gamma_i)$, together with general constraints on state and input (e.g., obstacle avoidance, kinematic limits).

B. Impulsive Impact Model

We consider impulsive impacts occurring at discrete instants. Let an impact occur at time \hat{t} . Stack the contact impulses as $\Phi^c = (\Phi_1^c, \dots, \Phi_{n^c}^c)$ with $\Phi_i^c = (\Phi_i^{c,t}, \Phi_i^{c,n})$, and let Φ^j denote the joint reaction impulse. The impact dynamics are

$$M(q(\hat{t}))(v(\hat{t}^+) - v(\hat{t}^-)) = J^c(q(\hat{t}))^\top R^c(q(\hat{t}))\Phi^c + \nabla c^j(q(\hat{t}))^\top \Phi^j, \quad (6a)$$

$$0 \leq \Phi_i^{c,n} \perp \text{sd}(p_i^c(q(\hat{t}))) \geq 0, \quad (6b)$$

$$0 = \Phi_i^{c,n} n_i^c(q(\hat{t}))^\top J_i^c(q(\hat{t}))(v(\hat{t}^+) + \varepsilon_i^c v(\hat{t}^-)), \quad (6c)$$

$$0 = \Phi_i^{c,n} (v_i^{c,t}(q(\hat{t}), v(\hat{t}^+)) + \Gamma_i \nabla \|\Phi_i^{c,t}\|_\sim^\top), \quad (6d)$$

$$0 \leq \Gamma_i \perp |\mu_i \Phi_i^{c,n}|_\sim - \|\Phi_i^{c,t}\|_\sim \geq 0, \quad (6e)$$

$$i = 1, \dots, n^c.$$

Conditions (6d)–(6e) play the same role as (4). They determine the tangential contact impulse via a stationarity condition (multiplied by normal contact impulse) and the friction cone constraint, with Γ_i the associated Lagrange multiplier. Impact elasticity is governed by (6c), where ε_i^c is the coefficient of restitution at the i th contact.

We remove the explicit joint reaction impulse in (6a) by imposing the joint constraint post impact at the velocity level, $\nabla c^j(q(\hat{t}^+))v(\hat{t}^+) = 0_{n^j}$. Solving the resulting constraint and (6a) yields an explicit expression for Φ^j , which we substitute back into (6a).

III. INTEGRAL CROSS COMPLEMENTARITY

We seek a solution to (1) and (6) over a time horizon $[0, t^f]$. We discretize $[0, t^f]$ into a grid of N nodes, $0 = t_1, \dots, t_N = t^f$, and we isolate all contact mode switches to occur only at the grid nodes. These contact modes indicate either no contact, sticking contact or slipping contact.

First, we introduce the following continuous-time cross-complementarity condition. Let $a, b : [0, t^f] \rightarrow \mathbb{R}_+$ be piecewise continuous with any discontinuities confined to the grid. Specifically, for each $k = 1, \dots, N-1$, a and b are continuous on $(t_k, t_{k+1}]$. They satisfy pointwise-in-time complementarity

$$0 \leq a(t) \perp b(t) \geq 0, \quad \forall t \in [0, t^f]. \quad (7)$$

To ensure switches between $a > 0, b = 0$ and $a = 0, b > 0$ happen at the grid nodes, we impose, for each $k = 1, \dots, N-1$,

$$0 \leq a(t) \perp b(t') \geq 0, \quad \forall t, t' \in (t_k, t_{k+1}], \quad (8)$$

together with $0 \leq a(t_1) \perp b(t_1) \geq 0$. We then have the following continuous-time analogue of [12, Lem. 2].

Lemma 1. *Let (8) hold and fix $k \in \{1, \dots, N-1\}$. If $a(t) > 0$ for some $t \in (t_k, t_{k+1}]$, then $b(t') = 0$ for all $t' \in (t_k, t_{k+1}]$. Conversely, if $b(t') > 0$ for some $t' \in (t_k, t_{k+1}]$, then $a(t) = 0$ for all $t \in (t_k, t_{k+1}]$.*

Proof. If $a(t) > 0$ at some $t \in (t_k, t_{k+1}]$ and $b(t') > 0$ at some $t' \in (t_k, t_{k+1}]$, then (8) is violated, which is a contradiction. \square

While (8) localizes switches to the grid nodes (a desirable property for both numerical reliability and physical realism), it is still a pointwise-in-time constraint and thus numerically intractable. Existing approaches [6, 12, 13] enforce discrete-time surrogates of (8) in aggregated forms, which risks inter-sample violations. To avoid this, we propose an integral reformulation, referred to as an *integral cross-complementarity* condition.

Proposition 1. For each $k = 1, \dots, N-1$, the constraint

$$0 \leq \left(\int_{t_k}^{t_{k+1}} a(t) dt \right) \perp \left(\int_{t_k}^{t_{k+1}} b(t) dt \right) \geq 0, \quad (9)$$

is equivalent to (8).

Proof. (\Rightarrow) If (9) holds and $\int_{t_k}^{t_{k+1}} a(t) dt > 0$, then the complementarity of the integrals implies $\int_{t_k}^{t_{k+1}} b(t) dt = 0$. Since a and b are nonnegative and are continuous on $(t_k, t_{k+1}]$, it follows that $a(t) > 0$ and $b(t) = 0$ for all $t \in (t_k, t_{k+1}]$. The argument is symmetric if $\int_{t_k}^{t_{k+1}} b(t) dt > 0$. Therefore, (8) holds.

(\Leftarrow) If (8) holds, Lemma 1 implies that for each $k = 1, \dots, N-1$, either $a(t) = 0$ or $b(t) = 0$ (or both) for all $t \in (t_k, t_{k+1}]$. Hence one (or both) of the integrals is zero, yielding (9). \square

In practice, we compute the integrals in (9) by integrating an auxiliary dynamical system

$$\dot{y} = (\dot{y}_{[1]}, \dot{y}_{[2]}) = (a, b), \quad (10)$$

and then impose the complementarity constraint

$$0 \leq y_{[1]}(t_{k+1}) - y_{[1]}(t_k) \perp y_{[2]}(t_{k+1}) - y_{[2]}(t_k) \geq 0. \quad (11)$$

If a or b are not inherently nonnegative, we enforce their nonnegativity through the path constraint function g . In (1), to ensure that contact mode switches occur only at the grid nodes, we impose integral cross-complementarity between the following pairs: (i) $\phi_i^{c,n}$ and $\text{sd}(p_i^c(q))$, (ii) $\phi_i^{c,n}$ and $\lambda_i^c(q, v, \phi_i^c, \gamma_i)$, and (iii) γ_i and $\rho_i^c(\phi_i^c, \gamma_i)$.

Remark 1. *Discrete-time cross-complementarity formulations (e.g., FESD-J [12]) typically introduce additional cross-complementarity constraints to detect sign changes in the tangential velocity at each contact point. Our formulation obviates these by parameterizing the control input u on each interval $(t_k, t_{k+1}]$, $k = 1, \dots, N-1$, with C^1 basis functions. Consequently, λ_i^c is also C^1 on $(t_k, t_{k+1}]$, which precludes nonsmoothness of the tangential velocity between nodes.*

IV. CONTACT-IMPLICIT TRAJECTORY OPTIMIZATION PROBLEM

In this section, we formulate the CITO problem for the maximal-coordinate model of Section II. The key steps are: (i) constructing an auxiliary system to handle path constraints and integral cross-complementarity, (ii) forming the time-dilated augmented system, (iii) parameterizing the control input and the dilation factor, and (iv) applying a smooth

relaxation of the complementarity constraints. We employ time-dilation [5, 15] so that the time-discretization associated with integral cross complementarity does not constrain the contact mode sequence (e.g., gait cadence and pattern).

A. Auxiliary System

We first construct the auxiliary system, which serves two purposes: (i) encoding cumulative violation of path constraints [5, Lem. 2], and (ii) representing the integral cross-complementarity constraints (see (10)). The auxiliary dynamics are

$$\dot{y} = \begin{bmatrix} \vdots \\ \dot{y}_{[4(i-1)+j]} \\ \vdots \\ \dots \\ \vdots \\ \dot{y}_{[4n^c+k]} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \phi_i^{c,n} \\ \lambda_i^c(q, v, \phi_i^{c,n}, \gamma_i) \\ \gamma_i \\ \rho_i^c(\phi_i^c, \gamma_i) \\ \vdots \\ \dots \\ M^g |g(x, u)|_{\tilde{\tau}} \end{bmatrix} = \Omega(x, u), \quad (12)$$

where $i = 1, \dots, n^c$, $j = 1, \dots, 4$, $k = 1, \dots, N^g$, and $M^g \in \mathbb{R}^{N^g \times n^g}$ is a nonnegative ‘‘mixing’’ matrix that allocates the n^g components of $|g(x, u)|_{\tilde{\tau}}$ across N^g components of the auxiliary state. Let $n^y = 4n^c + N^g$ denote the dimension of the auxiliary state. Continuous-time satisfaction of the path constraints on each grid interval $(t_k, t_{k+1}]$ is enforced by $y_{[4n^c+i]}(t_{k+1}) = y_{[4n^c+i]}(t_k)$ for $i = 1, \dots, N^g$ and $k = 1, \dots, N-1$. To preserve constraint qualification, we relax this to

$$y_{[4n^c+i]}(t_{k+1}) - y_{[4n^c+i]}(t_k) \leq \epsilon, \quad (13)$$

with a positive tolerance ϵ . With appropriate scaling and preconditioning of the resulting CITO problem, ϵ can be chosen as a small but numerically meaningful value (e.g., $\sim 10^{-4}$), which leads to negligible continuous-time constraint violation. See [5] for further discussion and analysis.

B. Parameterized Time-Dilated Augmented System

We define the time-dilated augmented system on the normalized interval $[0, 1]$ with augmented state $\tilde{x} = (x, y)$ and augmented control $\tilde{u} = (u, s)$

$$\dot{\tilde{x}} = s(f(x, u), \Omega(x, u)) = \tilde{f}(\tilde{x}, \tilde{u}), \quad (14)$$

where $\dot{\square} = d\square/d\tau$ for $\tau \in [0, 1]$, and $s = dt/d\tau$ is the dilation factor, treated as a control input. Let $n^{\tilde{x}} = n^x + n^y$ and $n^{\tilde{u}} = n^u + 1$.

We form an integral representation of (14) over each grid interval from Section III. First, for each $k = 1, \dots, N-1$, we map $[0, 1]$ to $[t_k, t_{k+1}]$, and parameterize the augmented control on $[0, 1]$ as

$$\tilde{u}(\tau) = (\nu(\tau)U, S),$$

where $\nu : [0, 1] \rightarrow \mathbb{R}^{n^u \times n^u N^u}$ is a matrix-valued polynomial basis function, and $U \in \mathbb{R}^{n^u N^u}$ and $S \in \mathbb{R}$ are the corresponding coefficients. Then the integral form of (14) over $[0, 1]$ is

$$\tilde{F}(\tilde{x}^-, U, S) = \tilde{x}^- + \int_0^1 \tilde{f}(\tilde{x}(\tau), (\nu(\tau)U, S)) d\tau, \quad (15)$$

for $\tilde{x}^- \in \mathbb{R}^{n^{\bar{x}}}$, $U \in \mathbb{R}^{n^u N^u}$, and $S \in \mathbb{R}$.

C. Nonconvex Optimization Problem

We can now cast the CITO problem as a nonconvex optimization problem. A key aspect of most existing approaches for CITO is a smooth relaxation of the complementarity constraints. Complementarity is inherently nonsmooth: for scalars a and b the condition $0 \leq a \perp b \geq 0$ is equivalent to $\min(a, b) = 0$, i.e., feasible set is the nonnegative coordinate axes. We employ the well-known \mathcal{C}^1 Fischer–Burmeister function to relax a complementarity constraint

$$\text{FB}^\delta(a, b) \leq 0, \quad a \geq 0, \quad b \geq 0, \quad (16)$$

where $\text{FB}^\delta(a, b) = a + b - \sqrt{a^2 + b^2 + \delta}$ and $\delta > 0$ controls the tightness of the relaxation. As $\delta \rightarrow 0$ the feasible set of (16) approaches that of $\min(a, b) = 0$. Note, any other \mathcal{C}^1 nonlinear complementarity function could be used in place of Fischer–Burmeister.

The decision variables of the CITO problem are: (i) pre- and post-impact augmented states at the grid nodes $\tilde{x}_1^-, \tilde{x}_1^+, \dots, \tilde{x}_N^-, \tilde{x}_N^+$, (ii) coefficients parameterizing the augmented control input on each grid interval $U_1, S_1, \dots, U_{N-1}, S_{N-1}$, and (iii) contact impulses and associated multipliers at the nodes $\Phi_1^c, \Gamma_1, \dots, \Phi_N^c, \Gamma_N$. Here $\tilde{x}_k^\square = (x_k^\square, y_k^\square)$ with $x_k^\square = (q_k^\square, v_k^\square)$, $\Phi_k^c = (\Phi_{k,1}^c, \dots, \Phi_{k,n^c}^c)$ with $\Phi_{k,i}^c = (\Phi_{k,i}^{c,t}, \Phi_{k,i}^{c,n})$, and $\Gamma_k = (\Gamma_{k,1}, \dots, \Gamma_{k,n^c})$, for $\square \in \{+, -\}$, $k = 1, \dots, N$, and $i = 1, \dots, n^c$.

The CITO problem comprises of the following constraints. First, the augmented dynamics is enforced in integral form (15), for $k = 1, \dots, N-1$. Next, (6a) is imposed at each grid node to allow potential impacts, along with continuity of pose and auxiliary state across impact, via $\tilde{G}(\tilde{x}_k^+, \tilde{x}_k^-, \Phi_k^c) = 0_{n^{\bar{x}}}$, for $k = 1, \dots, N$. Next, we require the parameterization coefficients to lie in convex sets, i.e., $U_k \in \mathcal{U}$ and $S_k \in \mathcal{S}$, for $k = 1, \dots, N-1$. Set \mathcal{U} encodes joint torque bounds, nonnegativity of the multiplier γ , and the contact force friction cone. Set \mathcal{S} is a positive compact interval. Next, integral cross complementarity, relaxed via (16), and the relaxed path constraint satisfaction (13) are aggregated as $\Psi^{\delta, \epsilon}(y_{k+1}^-, y_k^+) \leq 0_{n^\Psi}$, for $k = 1, \dots, N-1$. Next, the conditions (6b)-(6e) are collected into $\Theta^\delta(x_k^+, x_k^-, \Phi_k^c, \Gamma_k) \leq 0_{n^\Theta}$ using relaxation (16). Note that (6c) and (6d) are also expressed as complementarity constraints. Finally, we constrain the initial and terminal states x_i and x_f .

The CITO problem is then given by

$$\begin{aligned} & \text{minimize} && \sum_{k=1}^{N-1} L_k(x_k^+, U_k, S_k) && (17a) \\ & \tilde{x}_k^+, \tilde{x}_k^-, U_k && && \\ & S_k, \Phi_k^c, \Gamma_k && && \end{aligned}$$

$$\text{subject to } \tilde{x}_{k+1}^- = \tilde{F}(\tilde{x}_k^+, U_k, S_k), \quad (17b)$$

$$\Psi^{\delta, \epsilon}(y_{k+1}^-, y_k^+) \leq 0_{n^\Psi}, \quad (17c)$$

$$U_k \in \mathcal{U}, \quad S_k \in \mathcal{S}, \quad (17d)$$

$$k = 1, \dots, N-1,$$

$$\tilde{G}(\tilde{x}_k^+, \tilde{x}_k^-, \Phi_k^c) = 0_{n^{\bar{x}}}, \quad (17e)$$

$$\Theta^\delta(x_k^+, x_k^-, \Phi_k^c, \Gamma_k) \leq 0_{n^\Theta}, \quad (17f)$$

$$k = 1, \dots, N,$$

$$x_1^- = x_i, \quad x_N^+ = x_f. \quad (17g)$$

The cost function (17a) can represent various objectives, such as control effort, reference tracking, or maneuver time.

V. SEQUENTIAL CONVEX PROGRAMMING WITH BACKTRACKING HOMOTOPY

To solve the nonconvex problem (17), we employ sequential convex programming (SCP), specifically the prox-linear method with exact penalization of nonconvex constraints [5, Sec. 4]. Each SCP iteration solves a convex subproblem in which the nonconvex constraints are linearized and penalized with an ℓ_1 (exact) penalty, the convex constraints from (17) are kept intact, and a quadratic proximal term penalizes deviation from the previous iterate. With suitable weights for the proximal and exact-penalty terms, the iterates converge to a stationary point of the penalized nonconvex problem. If that point is feasible for (17), it is a KKT point.

Applying SCP to problems with complementarity constraints requires special care. When using a relaxation such as (16), fixing the tightness parameter δ a priori often stalls the algorithm and prevents convergence. For reliable convergence, δ is typically driven toward zero via a homotopy over the SCP iterations [16, 17]. We propose a homotopy scheme for δ , embedded within SCP (referred to as embedded δ -homotopy) with backtracking, in which δ is tightened progressively but can be temporarily relaxed to allow larger steps and broader search directions before committing to a smaller value. Algorithm 1 describes the approach. Let z_j , δ_j , J_j^{px} , and J_j^{ep} denote the subproblem solution, the tightness parameter, the proximal term, and the exact-penalty term, respectively, at SCP iterate j . Inputs to the algorithm are the initial tightness parameter δ_0 , an initial guess z_0 , tolerances $\delta^{\min} (< \delta_0)$, ϵ^{px} , ϵ^{ep} , a shrinkage factor $\alpha \in (0, 1)$, and a stall horizon N^{stall} . The update rule for the tightness parameter has three parts: (i) if neither J_j^{px} nor J_j^{ep} has decreased relative to iterate $M = \max(j - N^{\text{stall}}, 1)$, set δ_{j+1} to the log-scale bisection between δ_j and δ_M , (ii) else, if neither term has decreased relative to the previous iterate, set $\delta_{j+1} = \delta_j$; (iii) otherwise, shrink the tightness parameter as $\delta_{j+1} = \alpha \delta_j$.

Algorithm 1 δ -Homotopy

Require: $z_0, \delta_0, \delta^{\min}, \epsilon^{\text{px}}, \epsilon^{\text{ep}}, \alpha, N^{\text{stall}}$

Ensure: z_j

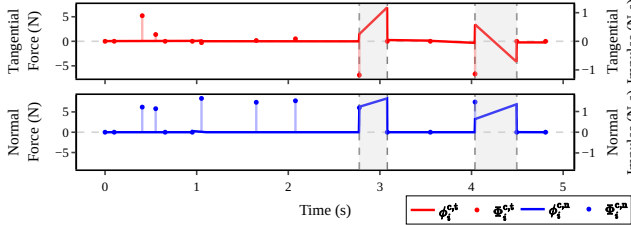
```

1:  $j \leftarrow 0, J_j^{\text{px}} \leftarrow 0, J_j^{\text{ep}} \leftarrow 0$ 
2: while  $J_j^{\text{px}} > \epsilon^{\text{px}} \vee J_j^{\text{ep}} > \epsilon^{\text{ep}} \vee \delta_j > \delta^{\min}$  do
3:    $z_{j+1}, J_{j+1}^{\text{px}}, J_{j+1}^{\text{ep}} \leftarrow \text{solve\_subproblem}(z_j, \delta_j)$ 
4:    $M \leftarrow \max(j - N^{\text{stall}}, 1)$ 
5:   if  $J_{j+1}^{\text{px}} > J_M^{\text{px}} \vee J_{j+1}^{\text{ep}} > J_M^{\text{ep}}$  then
6:      $\delta_{j+1} \leftarrow \exp(\frac{1}{2} \log(\delta_j \delta_M))$ 
7:   else if  $J_{j+1}^{\text{px}} > J_j^{\text{px}} \vee J_{j+1}^{\text{ep}} > J_j^{\text{ep}}$  then
8:      $\delta_{j+1} \leftarrow \delta_j$ 
9:   else
10:     $\delta_{j+1} \leftarrow \alpha \delta_j$ 
11:   end if
12:    $j \leftarrow j + 1$ 
13: end while

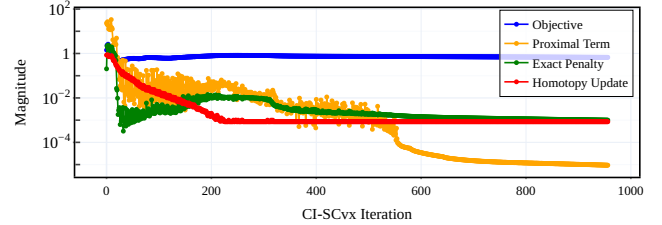
```



(a) Gait produced by a monopod locomotion solution. The path of the center of mass (CoM) of the upper leg is shown by the orange trace, and the path of the contact point (end of the lower leg) is shown by the cyan trace. The pose at grid nodes are displayed with higher opacity.



(b) Time histories of the normal and tangential contact forces and impulses. Grid intervals with sustained contact are denoted by gray regions.



(c) History of convergence metrics across CI-SCvx iterations, including the subproblem objective terms and homotopy tightness parameter δ_j .

Fig. 2: Monopod locomotion solution on a flat surface, produced by CI-SCvx.

VI. EXPERIMENTAL RESULTS

In this section, we detail the implementation of the proposed CI-SCvx framework and present case studies on trajectory optimization for monopedal and bipedal multi-body robots, illustrating the versatility of CI-SCvx. A notable advantage of our algorithm and formulation is its suitability for GPU-parallelized execution. The key computational task in solving (17) is transcription, i.e., constructing and linearizing (17b) and (17e), which define the hybrid dynamical model due to contact and impact. We carry out transcription on the GPU using a JAX backend by exploiting two features of the formulation: (i) multiple shooting enables parallelization across grid intervals, and (ii) the maximal-coordinate representation enables parallel evaluation of model elements (joint constraints, contact Jacobians, etc.). Transcription on each grid interval uses Internal Numerical Differentiation (IND) [18] with a custom Runge-Kutta-Fehlberg integrator and a first-order-hold (FOH) control input parameterization that is allowed to be discontinuous at grid nodes. Although more advanced JAX-compatible integrators (e.g., Diffrax) are available, in our experiments they were slower and did not improve SCP convergence. Beyond transcription, the remaining nonconvex constraints in (17) are also linearized on the GPU. All linearized quantities are transferred from the GPU to the CPU and assembled, together with the convex constraints of (17), into a canonical-form second-order cone program (SOCP) using compressed sparse column (CSC) format for the constraint and objective matrices. This process is referred to as canonicalization. During assembly, we apply row-wise normalization (preconditioning) to all constraint matrices and affine scaling to the decision variables, which improves solver reliability and the practical convergence of SCP. The resulting sparse data are passed directly to the QOCO convex solver [19], allowing us to bypass general-purpose modeling tools (e.g., CVXPY) and produce high-

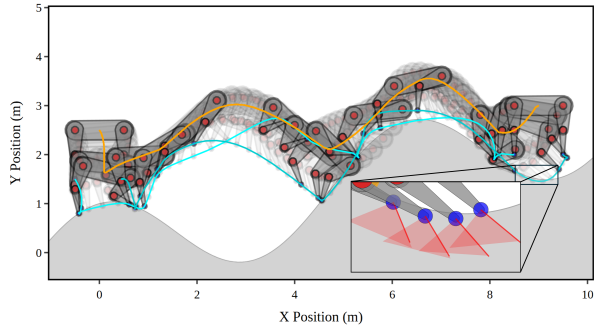
fidelity solutions in near real time (tens of seconds). Figure 1 illustrates the full CI-SCvx architecture.

All results are initialized by linearly interpolating poses at boundary conditions, setting velocities and joint torques to zero, and randomly sampling contact forces/impulses from the friction cone, which demonstrates that realistic trajectories can be obtained from naive initializations. A grid size of $N = 15$ and integration step count of 10 (per each grid interval) is chosen for all cases, demonstrating that we can resolve complex motions under sparse discretization. Each scenario minimizes the cumulative squared joint torques, with proximal and exact-penalty weights in the SCP subproblems set to 2000 and 50,000, respectively. We also globally set $\delta_0 = 1$, $\delta^{\min} = 10^{-3}$, $N^{\text{stall}} = 10$, $\alpha = 0.85$, $\mu_i = 1$ and $\epsilon_i^c = 0$, for $i = 1, \dots, n^c$. All results are visualized with the Plotly library in Python and forward-simulated with time steps of 0.01 s.

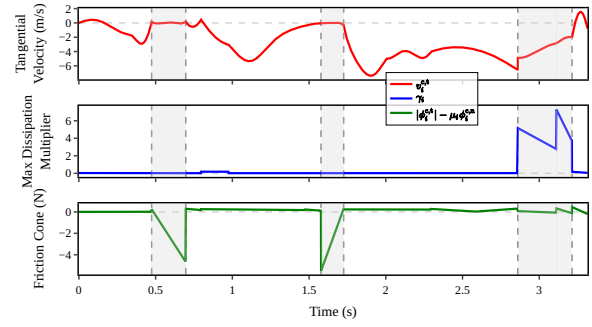
A. Monoped on a Flat Surface

We consider locomotion for a simple two-link monopod (Fig. 2) on a flat contact surface. Fig. 2a shows a largely periodic gait in which contact is predominantly instantaneous (impulsive) over the horizon. The corresponding impulses (Fig. 2b) are nonzero over most of the trajectory, while contact forces remain zero, yielding an energy-efficient “touch-and-go” motion driven by torque minimization. Although impacts are inelastic, an apparent “bouncing” motion is produced due to the use of joint torques. Sustained contact (gray window) occurs only in two intervals at the lift-off and landing points for the final lateral jump, at which point both the contact forces and impulses are activated. During this jump, the lower leg of the monopod lifts close to the upper leg to shift the system mass distribution.

Convergence metrics for CI-SCvx are shown in Fig. 2c. The tightness parameter δ decreases steadily with minor



(a) Gait produced by a biped locomotion solution. The path of the torso CoM is shown by the orange trace, and the paths of the contact points (front and hind feet) are shown by the cyan traces. The zoomed view shows slipping with contact forces activating the friction cone.



(b) Time histories of tangential velocity at contact point, max dissipation Lagrange multiplier and the friction cone term for the front foot, showing transitions between stick and slip motion. Impulses are not displayed as they are approximately zero for this scenario.

Fig. 3: Biped locomotion solution on a sinusoidal surface, produced by ci-SCvx .

oscillations due to the backtracking rule in Algorithm 1. Solutions are obtained in roughly 40 s on average, over about 800 ci-SCvx iterations. We apply the termination criteria on scaled variables, which results in high-accuracy solutions. Lower-accuracy yet physically reasonable trajectories for closed-loop tracking can be generated more quickly.

B. Biped on a Sinusoidal Surface

Our second case study considers a five-link biped traversing a sinusoidal surface (Fig. 3). We approximate the signed distance of contact point to the contact surface as

$$\text{sd}(p_i^c(q)) \approx \frac{h^c(p_i^c(q))}{\|\nabla h^c(p_i^c(q))\|}. \quad (18)$$

This choice (i) becomes arbitrarily accurate as $h^c(p_i^c(q)) \rightarrow 0$, (ii) is numerically stable far from the contact surface, and (iii) avoids solving a nonlinear optimization problem. This scenario showcases the capability of ci-SCvx to produce solutions with stick–slip transitions. The resulting gait (Fig. 3a) combines sticking and slipping along the uneven terrain to achieve feasible, efficient motion. In Fig. 3b, the tangential velocity is (approximately) zero during the first two persistent-contact phases (gray regions), indicating stick. In the third phase, slipping is activated over two grid intervals: the friction cone term approaches its boundary (i.e., $|\mu_i \phi_i^{c,n} - \|\phi_i^{c,t}\| \approx 0$), the maximum dissipation multiplier γ_i becomes nonzero, and the tangential velocity decreases toward zero. This behavior demonstrates that ci-SCvx can automatically determine the appropriate contact mode.

C. Demonstration with MuJoCo

The previous case studies demonstrated the expressivity of our formulation in simplified settings. Here we assess the applicability of ci-SCvx in a high-fidelity environment. Specifically, we use the Gymnasium HalfCheetah environment [9] with a MuJoCo backend, together with Google DeepMind’s MuJoCo MPC (MJPC) toolbox and its predictive sampling algorithm [20]. We employ MJPC for two purposes: (i) closed-loop tracking (with an LQR-style objective) of a given ci-SCvx trajectory, and (ii) a baseline controller optimized with a standalone reward which balances the default environment reward (maximize forward velocity, minimize normal-

ized joint torques) with a gait-phasing term that encourages alternating foot contacts. Sample rollouts are discarded if the torso exceeds a critical pitch angle or drops below a critical height (indicating tip-over and slumping failure modes).

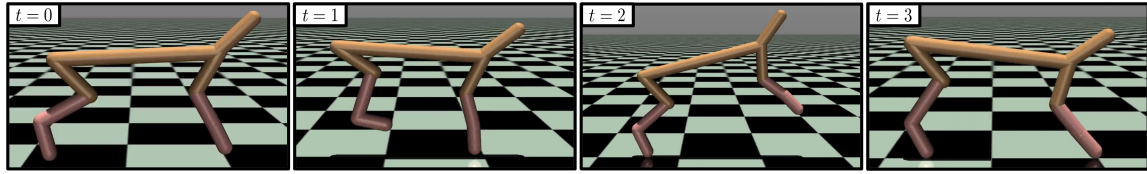
Fig. 4 compares an MJPC forward rollout using the standalone reward to an MJPC rollout that tracks a ci-SCvx solution, both on a horizon of $t^f = 3.2$ s. The ci-SCvx -tracked rollout requires less reward shaping than the standalone MJPC baseline and still achieves a 6.64% improvement in terms of the default environment objective as well as a 10.52% improvement in the cumulative joint torque cost. Qualitatively, the ci-SCvx trajectory yields a more stable gait, as illustrated in Fig. 4c.

D. Timing Comparisons to Existing SCP Implementations

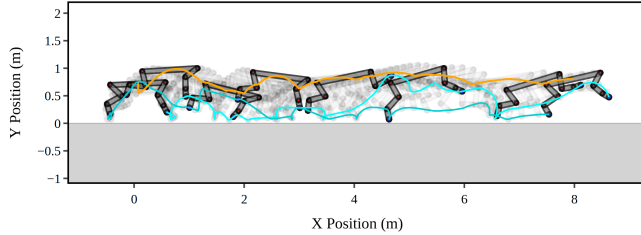
Finally, we compare the fully GPU-accelerated, customized version of ci-SCvx described in Section VI, which we refer to as ci-SCvx-CUSTOM , against a baseline version (ci-SCvx-BASE), which inherits from implementation approaches in OpenSCvx v0.2.0 [21], a state-of-the-art SCP toolbox for trajectory optimization. Specifically, we compare against OpenSCvx’s GPU parallelization and convex subproblem parsing pipeline (which relies on CVXPY). As shown in Fig. 5, ci-SCvx-CUSTOM achieves speedups of approximately $6\times$ (monoped) and $19\times$ (biped). Notably, CVXPY fails to parse the subproblem for models more complex than the biped. The primary advantages of ci-SCvx-CUSTOM relative to ci-SCvx-BASE arise from (i) an efficient JAX-based implementation for transcription and nonconvex constraint linearization, and (ii) direct canonicalization of subproblem that bypasses general-purpose parsers such as CVXPY.

VII. CONCLUSION

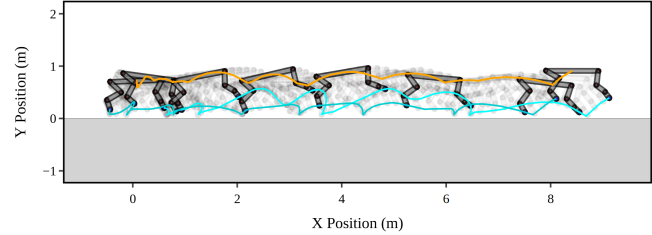
In this work, we developed a successive convexification-based framework for contact-implicit trajectory optimization (CITO), ci-SCvx , which introduces integral cross-complementarity constraints to obtain high-fidelity solutions with sparse time discretization. The approach also employs a backtracking homotopy scheme to ensure reliable convergence. We demonstrated ci-SCvx on complex locomotion tasks, showing performance and solve-time improvements



(a) Snapshots of the MJPC-tracked ci-SCvx solution visualized with Gymnasium.



(b) Gait produced by a standalone MJPC forward rollout.



(c) Gait produced by MJPC tracking of a ci-SCvx solution.

Fig. 4: Comparison between solutions produced for the Gymnasium HalfCheetah environment. We note that the “head” of the model is not visualized in (b)-(c) as it is massless and does not contribute to the model dynamics.

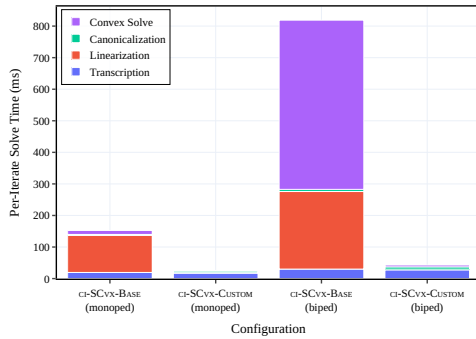


Fig. 5: Timing results comparing the proposed ci-SCvx-CUSTOM implementation against the baseline ci-SCvx-BASE.

over baselines in the literature. Future work will refine the homotopy strategy and solver implementation to reduce runtime and expand benchmarking to a broader set of models and trajectory optimization solvers.

REFERENCES

- [1] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. D. Prete. “Optimization-Based Control for Dynamic Legged Robots”. *IEEE Transactions on Robotics* 40 (2024), pp. 43–63.
- [2] A. Patel, S. L. Shield, S. Kazi, A. M. Johnson, and L. T. Biegler. “Contact-Implicit Trajectory Optimization Using Orthogonal Collocation”. *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 2242–2249.
- [3] A. Ö. Öno, P. Long, and T. Padir. “Contact-implicit trajectory optimization based on a variable smooth contact model and successive convexification”. *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2447–2453.
- [4] A. Ö. Öno, R. Corcodel, P. Long, and T. Padir. “Tuning-free contact-implicit trajectory optimization”. *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1183–1189.
- [5] P. Elango, D. Luo, A. G. Kamath, S. Uzun, T. Kim, and B. Açıkmeşe. “Continuous-time successive convexification for constrained trajectory optimization”. *Automatica* 180 (2025), p. 112464.
- [6] B. Baumrucker and L. Biegler. “MPEC strategies for optimization of a class of hybrid dynamic systems”. *Journal of Process Control* 19.8 (2009), pp. 1248–1256.
- [7] D. E. Orin, A. Goswami, and S.-H. Lee. “Centroidal dynamics of a humanoid robot”. *Autonomous Robots* 35.2–3 (2013), pp. 161–176.
- [8] D. E. Stewart. *Dynamics with Inequalities: Impacts and Hard Constraints*. Vol. 123. Other Titles in Applied Mathematics. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2011.
- [9] M. Towers, A. Kwiatkowski, J. Terry, J. U. Balis, G. De Cola, T. Deleu, M. Goulão, A. Kallinteris, M. Krimmel, A. KG, et al. “Gymnasium: A standard interface for reinforcement learning environments”. *arXiv preprint arXiv:2407.17032* (2024).
- [10] M. Posa, C. Cantu, and R. Tedrake. “A direct method for trajectory optimization of rigid bodies through contact”. *The International Journal of Robotics Research* 33.1 (2013), pp. 69–81.
- [11] I. Mordatch, E. Todorov, and Z. Popović. “Discovery of complex behaviors through contact-invariant optimization”. *ACM Transactions on Graphics* 31.4 (2012), pp. 1–8.
- [12] A. Nurkanović, J. Frey, A. Pozharskiy, and M. Diehl. “FESD-J: Finite Elements with Switch Detection for numerical optimal control of rigid bodies with impacts and Coulomb friction”. *Nonlinear Analysis: Hybrid Systems* 52 (2024), p. 101460.
- [13] S. R. Kazi, K. Wang, and L. Biegler. “On optimal control of hybrid dynamical systems using complementarity constraints”. *Journal of Process Control* 153 (2025), p. 103492.
- [14] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli. “Gait and Trajectory Optimization for Legged Systems Through Phase-Based End-Effector Parameterization”. *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 1560–1567.
- [15] A. G. Kamath, P. Elango, Y. Yu, S. Mceowen, G. M. Chari, J. M. Carson III, and B. Açıkmeşe. “Real-Time Sequential Conic Optimization for Multi-Phase Rocket Landing Guidance”. *IFAC-PapersOnLine* 56.2 (2023), pp. 3118–3125.
- [16] D. Malyuta and B. Açıkmeşe. “Fast homotopy for spacecraft rendezvous trajectory optimization with discrete logic”. *Journal of Guidance, Control, and Dynamics* 46.7 (2023), pp. 1262–1279.
- [17] J. Hall, A. Nurkanović, F. Messerer, and M. Diehl. “A sequential convex programming approach to solving quadratic programs and optimal control problems with linear complementarity constraints”. *IEEE Control Systems Letters* 6 (2021), pp. 536–541.
- [18] R. Quirynen. “Numerical simulation methods for embedded optimization”. PhD thesis. KU Leuven, 2017.
- [19] G. M. Chari and B. Açıkmeşe. “QOCO: A quadratic objective conic optimizer with custom solver generation”. *arXiv preprint arXiv:2503.12658* (2025).
- [20] T. Howell, N. Gileadi, S. Tuniyasuvunakool, K. Zakka, T. Erez, and Y. Tassa. “Predictive sampling: Real-time behaviour synthesis with MuJoCo”. *arXiv preprint arXiv:2212.00541* (2022).
- [21] C. R. Hayner, J. M. Carson III, B. Açıkmeşe, and K. Leung. “Continuous-Time Line-of-Sight Constrained Trajectory Planning for 6-Degree of Freedom Systems”. *IEEE Robotics and Automation Letters* (2025).