The International Journal of
Robotics Research
1–14
© The Author(s) 2019
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0278364919849235
journals.sagepub.com/home/ijr
SAGE

Contact-implicit trajectory optimization using variational integrators

Zachary Manchester, Neel Doshi, Robert J Wood
and Scott Kuindersma

Abstract

Contact constraints arise naturally in many robot planning problems. In recent years, a variety of contact-implicit *trajectory optimization algorithms have been developed that avoid the pitfalls of mode pre-specification by simultaneously optimizing state, input, and contact force trajectories. However, their reliance on first-order integrators leads to a linear tradeoff between optimization problem size and plan accuracy. To address this limitation, we propose a new family of trajectory optimization algorithms that leverage ideas from discrete variational mechanics to derive higher-order generalizations of the classic time-stepping method of Stewart and Trinkle. By using these dynamics formulations as constraints in direct trajectory optimization algorithms, it is possible to perform contact-implicit trajectory optimization with significantly higher accuracy. For concreteness, we derive a second-order method and evaluate it using several simulated rigid-body systems, including an underactuated biped and a quadruped. In addition, we use this second-order method to plan locomotion trajectories for a complex quadrupedal microrobot. The planned trajectories are evaluated on the physical platform and result in a number of performance improvements.*

Keywords

Contact, discrete mechanics, motion planning, trajectory optimization, microrobots

1. Introduction

Trajectory optimization algorithms comprise a powerful collection of methods for planning motions of nonlinear dynamical systems (Betts, 1998). Generally speaking, these algorithms aim to find an input trajectory that minimizes a cost function subject to a set of constraints on the system's states and inputs. Trajectory optimization has a long history of successful application to systems with smooth dynamics. However, many robotic systems experience discontinuous frictional contact with the environment as an essential part of their routine operation. The non-smooth dynamics encountered in these situations pose significant challenges.

A popular approach for handling contact events is to use a hybrid system model in which discontinuities are enumerated explicitly (Posa et al., 2016). However, contact mode sequences must then be pre-specified by the user or generated by a higher-level heuristic planner. This approach can work quite well for systems with a small number of contacts (Buss et al., 2003; Mombaur, 2009; Remy, 2011; Schultz and Mombaur, 2010; Xi and Remy, 2014). Unfortunately, for more complex systems, the number of modes grows exponentially with the number of contact constraints, making mode sequence pre-specification impractical.

Recently, an alternative approach has emerged in which state, input, and contact force trajectories are optimized simultaneously (Mordatch et al., 2012; Posa et al., 2014; Tassa et al., 2012). These so-called *contact-implicit* trajectory optimization methods can synthesize motions without *a priori* specification of the contact mode sequence. However, current state-of-the-art algorithms rely on first-order discretizations of the dynamics constraints, severely limiting accuracy and closed-loop trajectory tracking performance (Posa et al., 2016; Xi and Remy, 2014).

To overcome the accuracy limitations of current algorithms, we propose a new family of *variational* contact-implicit methods that combine ideas from discrete variational mechanics with the complementarity formulation of rigid-body contact to achieve higher-order integration

School of Engineering and Applied Sciences, Harvard University, Stanford, CA, USA

Corresponding author:
Zachary Manchester, School of Engineering and Applied Sciences, Harvard University, Durand Building, 496 Lomita Mall, Stanford, CA 94305, USA.
Email: zacmanchester@stanford.edu

accuracy. For simplicity and concreteness, we provide an explicit derivation of a second-order method. However, the mathematical tools used are general, and can be applied to derive integrators of arbitrary order.

This paper builds on two previous conference publications by the current authors (Doshi et al., 2018; Manchester and Kuindersma, 2017). The major contributions over those previous papers are a comparison of the new second-order contact-implicit method with a first-order method similar to that developed by Posa et al. (2014), as well as a discussion of the computational cost of the new method and extensive timing results. The remainder of the paper is organized as follows: Section 2 provides a summary of work on trajectory optimization through contact. Next, Section 3 gives a brief review of some important concepts from both classical variational mechanics and discrete mechanics. In Section 4 we derive the new variational rigid-body time-stepping scheme, and in Section 5 a direct trajectory optimization algorithm is built around these dynamics. Several simulation examples that demonstrate the performance of the new algorithm are then presented in Section 6. We also evaluate the performance of the algorithm in hardware on a complex quadrupedal microrobot platform in Section 7. Finally, we summarize our findings in Section 8.

2. Related work

In spite of their limitations, hybrid trajectory optimization algorithms that rely on a pre-specified contact mode sequence have had a number of notable successes. For example, hybrid multiple-shooting algorithms have been used to find open-loop stable walking trajectories for a two-dimensional model of a humanoid (Mombaur, 2009) and to study the energetics of quadrupedal locomotion (Remy, 2011). Hybrid collocation methods with third-order integration accuracy have also been demonstrated on a full-body model of a humanoid (Posa et al., 2016).

Much of the recent work on contact-implicit methods has focused on approximation schemes to smooth discontinuities. Several authors have developed indirect trajectory optimization algorithms based on differential dynamic programming (DDP) (Mayne, 1966) that apply a smoothing function to the contact constraints (Tassa et al., 2012) or penalize constraint violations in the objective function (Neunert et al., 2016; Todorov, 2011). The penalty approach has also been applied in direct trajectory optimization methods (Mordatch et al., 2012). These algorithms have been used to plan motions for quadrupeds (Neunert et al., 2016) and simplified humanoids (Mordatch et al., 2012; Tassa et al., 2012; Todorov, 2011) in simulation.

Another method for handling contact in trajectory optimization algorithms is to use a spring–damper model to generate contact forces. Such approaches often require very large spring stiffness and damping constant values to achieve realistic behavior, necessitating very small step sizes (Stewart, 2000). However, Neunert et al. (2017) have reported positive results using a nonlinear spring–damper model as part of a DDP-based algorithm. Their work is particularly notable for its successful demonstration in hardware experiments on a quadrupedal robot.

The previous work most closely related to the present paper is that of Posa et al. (2014). Their algorithm attempts to accurately capture discontinuous rigid-body physics by relying on the "time-stepping" linear complementarity formulation of Stewart and Trinkle (Anitescu and Potra, 1997; Stewart and Trinkle, 1996). The essential idea behind time-stepping methods for simulating rigid-body dynamics is to apply a first-order semi-implicit Euler discretization to the dynamics,

$$\begin{aligned} M(q_k)(v_{k+1} - v_k) &= h\big(B(q_k)u_k + N(q_k)^{\mathrm{T}}\gamma_k - C(q_k, v_{k+1})\big) \\ q_{k+1} &= q_k + hv_{k+1} \end{aligned} \tag{1}$$

where k is a time index; $q \in \mathbb{R}^{N_q}$, $v \in \mathbb{R}^{N_v}$, $u \in \mathbb{R}^M$, and $\gamma \in \mathbb{R}^P$ are configurations, velocities, control inputs, and normal contact impulses acting over a timestep of length h, respectively; M is the system's mass matrix; B and N^{T} are the Jacobians mapping control inputs and normal contact forces into generalized coordinates; and C includes Coriolis and potential terms. We have temporarily ignored the tangential (friction) component of the contact force for clarity, but it is discussed extensively in Section 4.2. For the normal impulse, we have the following constraints:

$$\begin{aligned} \gamma_k &\geqslant 0 \\ \phi(q_{k+1}) &\geqslant 0 \\ \gamma_k \phi(q_{k+1}) &= 0 \end{aligned} \tag{2}$$

where $\phi(q)$ is a function that returns the signed distance between closest points on bodies.

In words, the conditions in Equation (2) specify that normal forces can only push bodies apart (not pull them together), that bodies cannot interpenetrate, and that contact forces can only be non-zero when bodies are in contact. The combination of (1) and (2) forms a linear complementarity problem (LCP) that can be solved efficiently (Anitescu and Potra, 1997). However, this formulation depends crucially on the particular choice of first-order discretization used in (1). Although it may be possible to apply a higher-order discretization scheme in an *ad hoc* way, it is not obvious how to do so while still satisfying the contact constraints. To overcome this limitation, the next few sections introduce a set of mathematical tools for systematically deriving time-stepping methods with any desired order of integration accuracy.

At the core of our approach is the use of variational integrators as dynamics constraints in trajectory optimization. While not previously associated with contact dynamics, this idea has been explored by Junge et al. (2005), and has been termed "discrete mechanics and optimal control" (DMOC).

3. Preliminaries

This section reviews some classical results from variational mechanics, as well as some more recent results from discrete mechanics.

3.1. Lagrange–D'Alembert principle

Our starting point is the Lagrange–D'Alembert principle, which is the integral form of D'Alembert's principle of virtual work (Marsden and Ratiu, 1999), and can also be thought of as a modification of Hamilton's principle of least action (Goldstein et al., 2001) to accommodate external forces:

$$\delta \int_{t_0}^{t_f} \mathcal{L}(q,\dot{q})\ dt + \int_{t_0}^{t_f} F \cdot \delta q\ dt = 0 \tag{3}$$

We use $\mathcal{L}$ to denote the system's Lagrangian, F to denote a generalized force, and δ to indicate a variation (Fréchet derivative) with respect to the trajectory $q(t)$. Equation (3) describes a boundary-value problem in which a trajectory $q(t)$ is sought given fixed end points $q(t_0)$ and $q(t_f)$.

We now review the steps used to derive the classical forced Euler–Lagrange equation from (3). Applying the chain rule results in,

$$\int_{t_0}^{t_f} (D_1\mathcal{L}(q,\dot{q}) \cdot \delta q + D_2\mathcal{L}(q,\dot{q}) \cdot \delta\dot{q})\ dt + \int_{t_0}^{t_f} F \cdot \delta q\ dt = 0 \tag{4}$$

where we have moved the variational derivative inside the integral (Marsden and Ratiu, 1999) and used the *slot derivative* D_i to indicate partial differentiation with respect to a function's ith argument. The next step is to eliminate $\delta\dot{q}$ by performing an integration by parts:

$$\begin{aligned}&\int_{t_0}^{t_f} (D_1\mathcal{L}(q,\dot{q}) - \frac{d}{dt}D_2\mathcal{L}(q,\dot{q}) + F) \cdot \delta q\ dt \\ &+ D_2\mathcal{L}(q(t_f),\dot{q}(t_f)) \cdot \delta q(t_f) - D_2\mathcal{L}(q(t_0),\dot{q}(t_0)) \cdot \delta q(t_0) = 0\end{aligned} \tag{5}$$

The fact that the end points $q(t_0)$ and $q(t_f)$ of the boundary value problem are fixed, and thus $\delta q(t_0) = \delta q(t_f) = 0$, can be used to eliminate the last two terms in (5):

$$\int_{t_0}^{t_f} \left(D_1\mathcal{L}(q,\dot{q}) - \frac{d}{dt}D_2\mathcal{L}(q,\dot{q}) + F \right) \cdot \delta q\ dt = 0 \tag{6}$$

Finally, recognizing that (6) must hold for all variations δq, we arrive at the classical forced Euler–Lagrange equation:

$$\frac{d}{dt}D_2\mathcal{L}(q,\dot{q}) - D_1\mathcal{L}(q,\dot{q}) = F \tag{7}$$

By substituting a Lagrangian of the form,

$$\mathcal{L}(q,\dot{q}) = \frac{1}{2}\dot{q}^{\mathrm{T}}M(q)\dot{q} - V(q) \tag{8}$$

into (7), where $M(q)$ is a positive-definite mass matrix and $V(q)$ is a potential energy function, the familiar manipulator equation can be recovered by a simple application of the chain rule,

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = F \tag{9}$$

where $C(q,\dot{q})$ includes Coriolis terms and $G(q) = \nabla V(q)$ accounts for conservative forces. Rather than discretizing (9) in time, as in most prior work, our approach builds on ideas from discrete mechanics.

3.2. Discrete mechanics

Discrete mechanics encompasses a set of mathematical tools for deriving specialized numerical integrators for mechanical systems. These so-called *variational integrators* have many advantages over traditional Runge–Kutta schemes, including realistic long-term energy and momentum behavior (Marsden and West, 2001). While often associated with simulations of energy-conserving systems, variational integrators can also be applied to non-conservative systems subject to external forces and control inputs (Junge et al., 2005).

Inspired by the numerical methods used to solve optimal control problems, the strategy behind discrete mechanics is to approximate the integrals in the Lagrange–D'Alembert principle (3) with a quadrature rule *before* taking variations. We begin by breaking those integrals into N smaller pieces,

$$\delta \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \mathcal{L}(q,\dot{q})\ dt + \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} F(q,\dot{q}) \cdot \delta q\ dt = 0 \tag{10}$$

where $t_k = t_0 + kh$ and h is a small timestep. Each short integral in (10) is then approximated. While any quadrature rule can be used for this purpose, we will use the midpoint rule for simplicity and clarity:

$$\begin{aligned}&\delta \sum_{k=0}^{N-1} h\mathcal{L}\Big(\frac{q_k + q_{k+1}}{2}, \frac{q_{k+1} - q_k}{h}\Big) \\ &+ \sum_{k=0}^{N-1} hF\Big(\frac{q_k + q_{k+1}}{2}, \frac{q_{k+1} - q_k}{h}\Big) \cdot \Big(\frac{\delta q_k + \delta q_{k+1}}{2}\Big) = 0\end{aligned} \tag{11}$$

Equation (11) can be written more compactly as,

$$\delta \sum_{k=0}^{N-1} \mathcal{L}_d(q_k, q_{k+1}) + \sum_{k=0}^{N-1} \frac{1}{2}F_d(q_k, q_{k+1}) \cdot (\delta q_k + \delta q_{k+1}) = 0 \tag{12}$$

where $\mathcal{L}_d$ is known as the *discrete Lagrangian*,

$$\mathcal{L}_d(q_k, q_{k+1}) = h\mathcal{L}\left(\frac{q_k + q_{k+1}}{2}, \frac{q_{k+1} - q_k}{h}\right) \tag{13}$$

and F_d is called the *discrete generalized force*,

$$F_d(q_k, q_{k+1}) = hF\left(\frac{q_k + q_{k+1}}{2}, \frac{q_{k+1} - q_k}{h}\right) \tag{14}$$

Note that both the discrete Lagrangian and discrete generalized force depend on our particular choice of quadrature rule and, as a result, many different definitions are possible (Marsden and West, 2001).

We now apply the chain rule to the variation in (12):

$$\begin{aligned}&\sum_{k=0}^{N-1}(D_1\mathcal{L}_d(q_k, q_{k+1}) \cdot \delta q_k + D_2\mathcal{L}_d(q_k, q_{k+1}) \cdot \delta q_{k+1})\\ &+ \sum_{k=0}^{N-1}\frac{1}{2}F_d(q_k, q_{k+1}) \cdot (\delta q_k + \delta q_{k+1}) = 0\end{aligned} \tag{15}$$

Paralleling the derivation of the classical Euler–Lagrange equation in the previous section, we perform the discrete-time equivalent of integration by parts to line up the time indices of the δq terms. This amounts to a simple index manipulation trick:

$$\begin{aligned}&D_1\mathcal{L}_d(q_0, q_1)\delta q_0 + \frac{1}{2}F_d(q_0, q_1) \cdot \delta q_0\\ &+ \sum_{k=1}^{N-1}(D_2\mathcal{L}_d(q_{k-1}, q_k) + D_1\mathcal{L}_d(q_k, q_{k+1}).\\ &+ \frac{1}{2}F_d(q_{k-1}, q_k) + \frac{1}{2}F_d(q_k, q_{k+1})\Big) \cdot \delta q_k\\ &+ D_2\mathcal{L}_d(q_{N-1}, q_N)\delta q_N + \frac{1}{2}F_d(q_{N-1}, q_N) \cdot \delta q_N = 0\end{aligned} \tag{16}$$

As in the continuous case, the endpoints q_0 and q_N are fixed. As a result, $\delta q_0 = \delta q_N = 0$, and the first and last terms in (16) can be eliminated:

$$\begin{aligned}&\sum_{k=1}^{N-1}(D_2\mathcal{L}_d(q_{k-1}, q_k) + D_1\mathcal{L}_d(q_k, q_{k+1}) + \frac{1}{2}F_d(q_{k-1}, q_k)\\ &+ \frac{1}{2}F_d(q_k, q_{k+1})) \cdot \delta q_k = 0\end{aligned} \tag{17}$$

Finally, using the fact that (17) must hold for all variations δq_k, we arrive at the following discrete-time version of the forced Euler–Lagrange equation:

$$\begin{aligned}&D_2\mathcal{L}_d(q_{k-1}, q_k) + D_1\mathcal{L}_d(q_k, q_{k+1})\\ &+ \frac{1}{2}F_d(q_{k-1}, q_k) + \frac{1}{2}F_d(q_k, q_{k+1}) = 0\end{aligned} \tag{18}$$

Equation (18) can be used to simulate the dynamics of a mechanical system by inserting values for q_{k-1} and q_k and solving for q_{k+1}. In fact, it is equivalent to the implicit midpoint method.

An important result in the theory of discrete mechanics is that the order of accuracy associated with a variational integrator is equal to the order of accuracy of the quadrature rule used to define the discrete Lagrangian and discrete generalized force (Marsden and West, 2001). As the midpoint rule has a global error of $O(h^2)$, an integrator based on (18) inherits this second-order accuracy. Variational integrators of any desired order can be derived by simply choosing an appropriate quadrature rule (Ober-Blobaum and Saake, 2015).

4. Variational time-stepping methods

Time-stepping methods for simulating rigid-body dynamics with contact were popularized by Stewart and Trinkle (1996). The essential idea is to deal with the discontinuities that occur during rigid-body impacts by formulating the dynamics at the level of impulses and velocities, rather than forces and accelerations. The contact impulse produced during a timestep, together with the next state, is computed by solving a constrained optimization problem.

Since variational integrators such as (18) are also formulated in terms of impulses and avoid direct computation of forces and accelerations, they are a natural choice for handling rigid-body contact dynamics. In this section, we derive a time-stepping method with second-order integration accuracy. We treat only the case of perfectly inelastic collisions, meaning that the normal component of the velocity at the point of contact is set to zero upon impact. However, an extension to the elastic case is possible using existing time-stepping schemes (Anitescu and Potra, 1997).

4.1. Interpenetration and complementarity

Interpenetration must not occur between rigid bodies. Mathematically, this constraint can be expressed as an inequality,

$$\phi(q) \geqslant 0 \tag{19}$$

where $\phi(q)$ is a vector-valued function that evaluates the signed distance between closest points on all pairs of bodies.

To build a variational integrator that respects the interpenetration constraint, we add it to the discrete Lagrange–D'Alembert principle (12) with a corresponding Lagrange multiplier γ:

$$\begin{aligned}&\delta\sum_{k=0}^{N-1}\mathcal{L}_d(q_k, q_{k+1}) + \gamma_k^{\mathrm{T}}\phi(q_{k+1})\\ &+ \sum_{k=0}^{N-1}\frac{1}{2}F_d(q_k, q_{k+1}) \cdot (\delta q_k + \delta q_{k+1}) = 0\end{aligned} \tag{20}$$

Note the deliberate choice of time indices in the constraint term $\gamma_k^{\mathrm{T}}\phi(q_{k+1})$ to indicate that the *next* state must always satisfy the inequality $\phi(q_{k+1}) \geqslant 0$.

Following the same steps used to derive equation (18) in the previous section, we find

$$D_2\mathcal{L}_d(q_{k-1},q_k)+D_1\mathcal{L}_d(q_k,q_{k+1})+\frac{1}{2}F_d(q_{k-1},q_k) \\ +\frac{1}{2}F_d(q_k,q_{k+1})+N(q_{k+1})^{\mathrm{T}}\gamma_k=0 \tag{21}$$

where $N(q)^{\mathrm{T}}=(\partial\phi/\partial q)^{\mathrm{T}}$ is the Jacobian mapping normal contact forces into generalized coordinates. In addition, solutions to (21) must satisfy the following conditions:

$$\begin{aligned} &\gamma_k \geqslant 0 \\ &\phi(q_{k+1}) \geqslant 0 \\ &\gamma_k^{\mathrm{T}}\phi(q_{k+1})=0 \end{aligned} \tag{22}$$

Together, (21) and (22) form the first-order necessary conditions, known as Karush–Kuhn–Tucker (KKT) conditions, for an inequality constrained optimization problem (Boyd and Vandenberghe, 2004).

Physically, the Lagrange multiplier, γ_k (units Ns), takes on the magnitude of the contact impulse in the normal direction. The three conditions in (22) are collectively known as a *complementarity constraint*. In addition to preventing interpenetration, they ensure that contact forces can only push bodies apart (not pull them together), and that contact forces can only act when bodies are in contact. Such constraints are commonly denoted using the following shorthand notation:

$$0 \leqslant \gamma_k \perp \phi(q_{k+1}) \geqslant 0 \tag{23}$$

Intuitively, complementarity constraints express discontinuous “switching” behavior: only one variable or the other is allowed to be non-zero at a time. They are an inherent feature in many models of contact physics.

4.2. Coulomb friction

Coulomb friction exerts forces in the plane tangent to the contact surface between two bodies. It can be described by the *maximum dissipation principle* (Moreau, 1973), which states that friction forces maximize the rate of dissipation of kinetic energy. Mathematically, this can be posed as the following optimization problem,

$$\begin{aligned} \underset{b}{\text{minimize}} \quad & \dot{q}^{\mathrm{T}}D^{\mathrm{T}}b \\ \text{subject to} \quad & ||b|| \leqslant \mu\gamma \end{aligned} \tag{24}$$

where b is the friction force in the contact tangent plane, μ is the friction coefficient, and D^{T} is the Jacobian mapping tangential contact forces into generalized forces.

The second-order cone constraint in (24) is known as the *friction cone*. Because it is not differentiable at $b=0$, this constraint can pose difficulties for general-purpose nonlinear optimization solvers (Vanderbei and Yurttan, 1998). For this reason, an inner approximation of the friction cone is often made using a polyhedron (Stewart and Trinkle, 1996). For the specific case of a four-sided pyramid, this is accomplished by defining a new friction vector β with twice as many elements as b, enforcing the constraint $\beta \geqslant 0$, and forming a new Jacobian matrix (Anitescu and Potra, 1997):

$$P=\begin{bmatrix} D \\ -D \end{bmatrix} \tag{25}$$

With this approximation, the optimization problem (24) becomes

$$\begin{aligned} \underset{\beta}{\text{minimize}} \quad & \dot{q}^{\mathrm{T}}P^{\mathrm{T}}\beta \\ \text{subject to} \quad & \mu\gamma - e^{\mathrm{T}}\beta \geqslant 0 \\ & \beta \geqslant 0 \end{aligned} \tag{26}$$

where e is a vector of ones.

The set of first-order necessary conditions (KKT conditions) for an optimum of (26) are

$$\begin{aligned} &P\dot{q}+\psi e-\eta=0 \\ &\beta,\psi,\eta \geqslant 0 \\ &\mu\gamma - e^{\mathrm{T}}\beta \geqslant 0 \\ &\psi^{\mathrm{T}}\left(\mu\gamma - e^{\mathrm{T}}\beta\right)=0 \\ &\eta^{\mathrm{T}}\beta=0 \end{aligned} \tag{27}$$

where ψ (units m/s) and η (units m/s) are Lagrange multipliers. In the more compact shorthand notation introduced in the previous subsection, these conditions can be rewritten as

$$\begin{aligned} &P\dot{q}+\psi e-\eta=0 \\ &0 \leqslant \psi \perp \left(\mu\gamma - e^{\mathrm{T}}\beta\right) \geqslant 0 \\ &0 \leqslant \eta \perp \beta \geqslant 0 \end{aligned} \tag{28}$$

Physically, the Lagrange multiplier ψ approximates the projection of the system’s velocity into the plane tangent to the contact manifold. The conditions in (28) ensure that the friction force will assume whatever value is necessary to prevent sliding when $\psi=0$, up to the boundary of the friction cone. In the sliding case, when $\psi \neq 0$, the friction force will lie on the boundary of the friction cone.

4.3. A second-order time-stepping method

We now build a complete time-stepping method by combining the results of the previous sections. We first define a vector, λ, that combines the normal and friction components of the contact impulse,

$$\lambda=\begin{bmatrix} \gamma \\ \beta \end{bmatrix} \tag{29}$$

and the corresponding Jacobian matrix to map λ into generalized coordinates:

$$J = \begin{bmatrix} N \\ P \end{bmatrix} \tag{30}$$

The discrete Euler–Lagrange dynamics can then be written as follows:

$$\begin{aligned} &D_2\mathcal{L}_d(q_{k-1}, q_k) + D_1\mathcal{L}_d(q_k, q_{k+1}) + \frac{1}{2}F_d(q_{k-1}, q_k) \\ &+ \frac{1}{2}F_d(q_k, q_{k+1}) + J(q_{k+1})^{\mathrm{T}}\lambda_k = 0 \end{aligned} \tag{31}$$

The set of complementarity conditions derived in the previous subsections are used to determine λ_k in (31). Given q_{k-1} and q_k, the following feasibility problem can be solved to find λ_k and q_{k+1},

$$\begin{aligned} &r(h, q_{k-1}, q_k, q_{k+1}, \lambda_k) = 0 \\ &P(q_{k+1})\left(\frac{q_{k+1} - q_k}{h}\right) + \psi_k e - \eta_k = 0 \\ &0 \leqslant (\mu\gamma_k - e^{\mathrm{T}}\beta_k) \perp \psi_k \geqslant 0 \\ &0 \leqslant \phi(q_{k+1}) \perp \gamma_k \geqslant 0 \\ &0 \leqslant \beta_k \perp \eta_k \geqslant 0 \end{aligned} \tag{32}$$

where $r(h, q_{k-1}, q_k, q_{k+1}, \lambda_k) = 0$ refers to (31).

5. Direct trajectory optimization

We now propose a direct trajectory optimization algorithm that uses the variational time-stepping scheme developed in the previous section as a set of dynamics constraints. Our strategy is to formulate the trajectory optimization problem as a nonlinear program (NLP) and solve it using standard constrained optimization software.

To ease the numerical difficulties associated with complementarity constraints, we apply a smoothing scheme similar to that used by Fletcher and Leyffer (2004). The key idea is to relax the equality constraints in the three complementarity conditions in (32) by replacing them with inequalities and introducing slack variables s_k:

$$\begin{aligned} &r(h, q_{k-1}, q_k, q_{k+1}, \lambda_k) = 0 \\ &P(q_{k+1})\left(\frac{q_{k+1} - q_k}{h}\right) + \psi_k e - \eta_k = 0 \\ &\lambda_k, \psi_k, \eta_k, s_k \geqslant 0 \\ &\phi(q_{k+1}) \geqslant 0 \\ &(\mu\gamma_k - e^{\mathrm{T}}\beta_k) \geqslant 0 \\ &s_k - \eta_k^{\mathrm{T}}\beta_k \geqslant 0 \\ &s_k - \gamma_k^{\mathrm{T}}\phi(q_{k+1}) \geqslant 0 \\ &s_k - \psi_k^{\mathrm{T}}(\mu\gamma_k - e^{\mathrm{T}}\beta_k) \geqslant 0 \end{aligned} \tag{33}$$

Figure 1 illustrates the feasible regions for both the original "strict" complementarity constraints and the new relaxed complementarity constraints in (33). If the slack variables are reduced to zero, the two regions coincide.

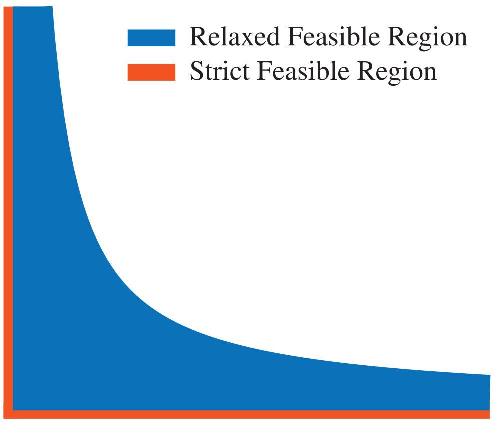

Fig. 1. Illustration of feasible regions for relaxed and strict complementarity constraints.

Physically, the relaxed complementarity constraints allow contact forces to act at a non-zero distance from the contact manifold. This aids numerical convergence, but we ultimately want solutions to closely respect the true constraints. To encourage convergence of solutions towards strict satisfaction of the original complementarity constraints, we augment the cost function with a term that penalizes s_k. The complete trajectory optimization problem can then be stated as the following NLP,

$$\begin{aligned} \underset{h, \mathcal{Q}, \mathcal{U}, \mathcal{C}}{\text{minimize}} \quad & J(h, \mathcal{Q}, \mathcal{U}) + \alpha \sum_{k=1}^{N-1} s_k \\ \text{subject to} \quad & f(h, q_{k-1}, q_k, q_{k+1}, \lambda_k, \psi_k, \eta_k) = 0 \\ & g(q_{k+1}, \lambda_k, \psi_k, \eta_k, s_k) \geqslant 0 \\ & u_{min} \leqslant u_k \leqslant u_{max} \\ & h_{\min} \leqslant h \leqslant h_{\max} \end{aligned} \tag{34}$$

where J is a cost function, α is a positive scalar weighting parameter, f and g are the equality and inequality constraints in (33), $\mathcal{Q}$ is the set of all configuration knot points, q_k, $\mathcal{U}$ is the set of all control inputs, u_k, and $\mathcal{C}$ is the set of all contact-related variables, λ_k, ψ_k, η_k, and s_k.

The penalty on the slack variables in the cost function of (34) is a so-called "exact penalty" that has theoretical convergence guarantees with finite values of α (Anitescu, 2005). In practice, we have observed good convergence behavior with modest values of α. Problem (34) can be solved with standard nonlinear programming algorithms such as sequential quadratic programming (SQP) and interior-point methods (Nocedal and Wright, 2006). We use the commercially available constrained optimization solver SNOPT (Gill et al., 2005). It is also straightforward to include additional constraints on the system's state and inputs.

6. Numerical examples

To evaluate the proposed trajectory optimization algorithm, we demonstrate its ability to generate complex, multi-contact motions by optimizing walking trajectories for two simulated legged robots: Spring Flamingo and LittleDog.

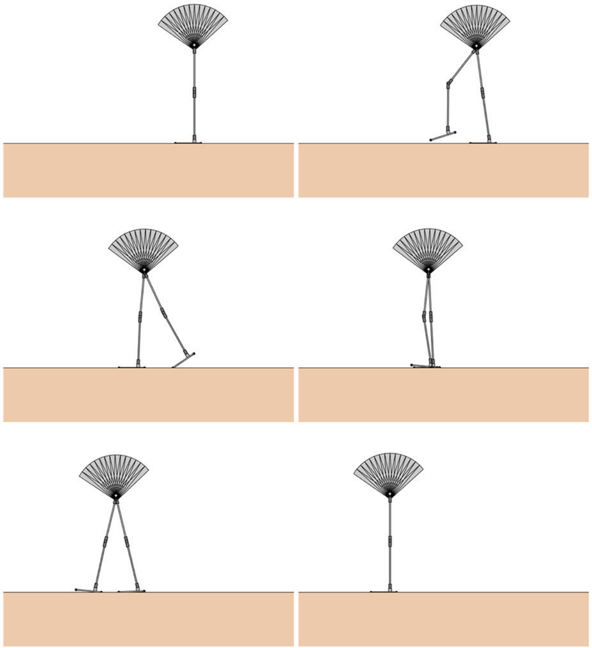

Fig. 2. Optimized walking gait for the Spring Flamingo robot.

Our implementation of this algorithm was written in MATLAB R2016a using the dynamics and control toolbox Drake written by Tedrake and the Drake Development Team (2016). We also compared its accuracy with the first-order method used by Posa et al. (2014) in both open- and closed-loop simulations. In all cases, the optimizer was initialized with dynamically infeasible trajectories consisting of simple linear interpolation between initial and goal states. No *a priori* information about contact forces or mode sequences was used.

6.1. Spring Flamingo

Spring Flamingo is an 18-state planar bipedal robot with actuated hips and knees and a passive spring ankle joint (Pratt and Pratt, 1998). A trajectory optimization problem was defined in which the robot was required to move from an initial standing pose to a final standing pose translated to the left. The following cost function was minimized,

$$J = \sum_{i=1}^{N-1} 0.1(x_i - x_g)^{\mathrm{T}}(x_i - x_g) + u_i^{\mathrm{T}} u_i \qquad (35)$$

where x_g is the goal state. Figure 2 shows a sequence of frames taken from the optimized walking gait. The algorithm produced an energetically efficient heel–toe gait that exploits the passive dynamics of the leg and ankle.

6.2. LittleDog

LittleDog is a 36-state quadrupedal robot designed by Boston Dynamics to enable research on legged locomotion (Buchli et al., 2010). A trajectory optimization problem was defined in which the robot was required to climb up an 11 cm step. Once again, initial and final state constraints were enforced and a simple quadratic cost function was minimized. Figure 3 shows an example climbing strategy and Figure 4 shows the corresponding sequence of modes (combinations of foot contacts) that were generated by the solver.

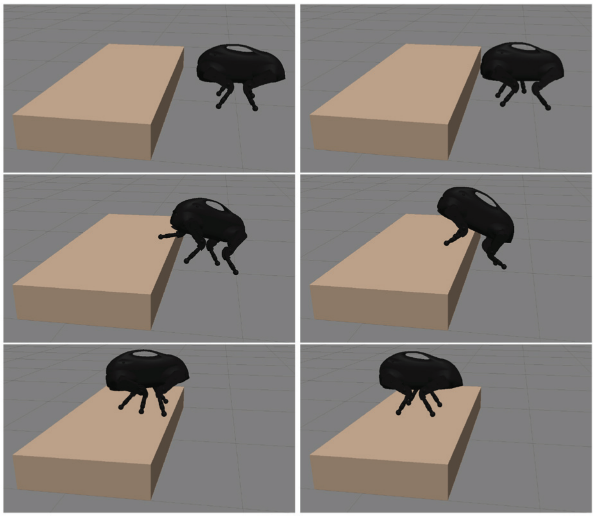

Fig. 3. LittleDog climbing a step.

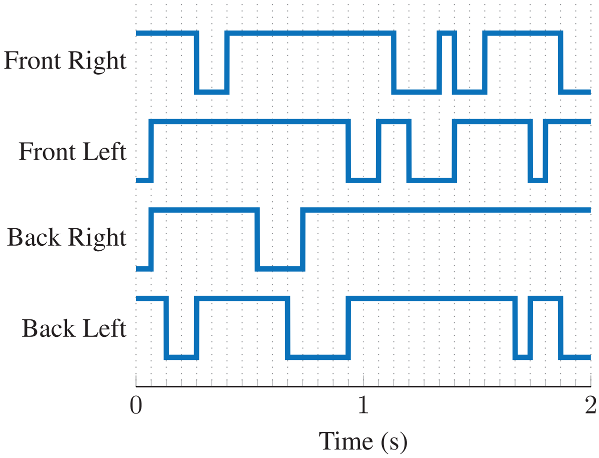

Fig. 4. Contact mode sequence for each foot from LittleDog step-climbing example.

6.3. Computational cost

We also compared the computational cost of our second-order method with a first-order variational method with the same order of accuracy as the time-stepping dynamics from Posa et al. (2014) for the Spring Flamingo and LittleDog examples. Our first-order method uses a backward-Euler discretization of the configurations and a midpoint

Table 1. Running time of first- and second-order methods.

Numerical example	First order (min)	Second order (min)
Spring Flamingo	4.30±1.72 ($n=5$)	4.91±2.17 ($n=5$)
LittleDog	38.77±4.50 ($n=5$)	52.34±6.36 ($n=5$)

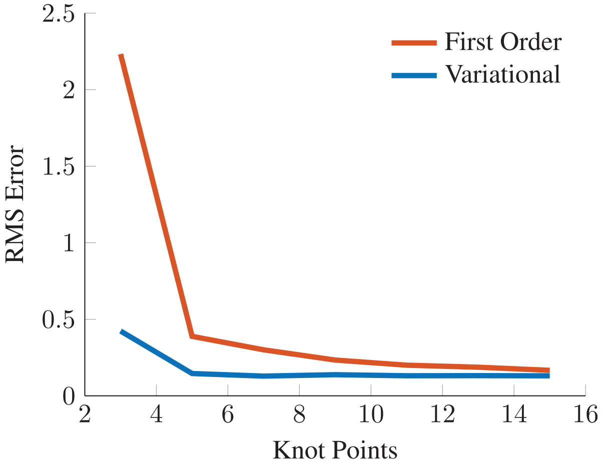

Fig. 5. Falling brick simulation RMS error and standard deviation.

discretization of the velocities. Five trajectory optimizations were run using the first- and second-order methods for each example on an Intel Core i5-6400 CPU with four 2.7 GHz cores and 8 GB of RAM. The mean running time to convergence ± one standard deviation is reported in Table 1.

Furthermore, the maximum slack variables for all 10 trials using the second-order method were less than the solver tolerances (10^{-4} for the Spring Flamingo and 10^{-5} for LittleDog) at convergence. This indicates that the solutions respect the true complementarity constraints and the second-order method does not return solutions in the relaxed feasibility region. On the other hand, the first-order method had difficulty finding a solution that respected the strict feasibility region for the LittleDog step-up example.

6.4. Simulation accuracy

Finally, we compare the first-order time-stepping dynamics used by Posa et al. (2014) with our second-order variational time-stepping method in open-loop simulations of a tumbling brick hitting the ground and closed-loop simulations of the LittleDog robot tracking a walking trajectory with proportional–derivative (PD) feedback control applied to its joints. Tumbling-brick simulations were initialized with twenty different randomly chosen initial conditions while varying the number of trajectory knot points. A reference solution was computed using the first-order method at a sample rate of 2 kHz. Figure 5 shows the root-mean-square (RMS) error (compared with the reference solution) as a function of the number of knot points. The variational method achieves better accuracy with fewer knot points (lower sampling rates) than the first-order time-stepping method.

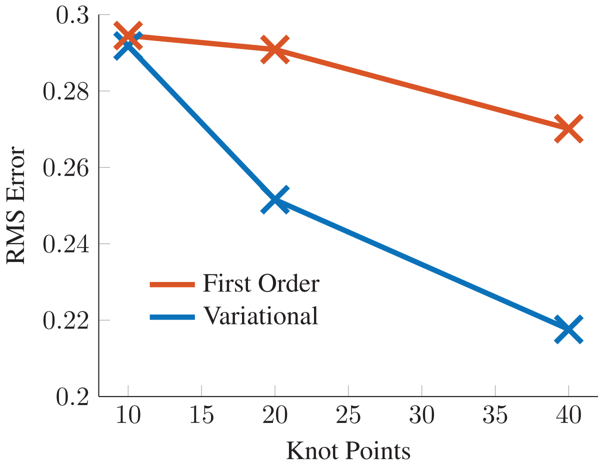

Fig. 6. RMS tracking error for LittleDog walking with PD tracking controller.

To test closed-loop tracking performance, a set of walking trajectories was optimized using both the first-order method of Posa et al. (2014) and the second-order variational method. The number of knot points used to parameterize the trajectories was varied between 10 and 40. Simple PD control was applied to each joint of the robot, and simulations were performed using the first-order method at a sample rate of 2 kHz. Figure 6 shows the RMS tracking error in the robot’s state.

The first- and second-order methods generate qualitatively similar plans and have similar running times. However, thanks to the improved accuracy of the second-order dynamics formulation, the plans generated using the second-order method achieve better closed-loop tracking performance with fewer knot points than the first-order method.

7. Hardware implementation

To demonstrate the practical utility of the new algorithm, its performance was evaluated on a quadrupedal microrobot, the Harvard Ambulatory MicroRobot (HAMR-VI, Figure 7). This robot is fabricated using laminate manufacturing techniques (Whitney et al., 2011), is 4.51 cm long, weighs 1.5 g, and has eight independently actuated degrees of freedom (Doshi et al., 2015). Each leg has two degrees of freedom that are driven by optimal-energy-density piezoelectric bending actuators (Jafferis et al., 2015). A flexure-based spherical-five-bar (SFB) transmission connects the two actuators to a single leg in a nominally decoupled manner: one actuator controls leg swing (x-direction) motion, while the other actuator controls leg lift (z-direction) motion.

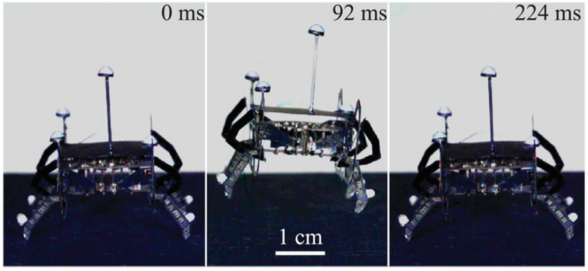

Fig. 7. Still frames of HAMR-VI executing a vertical jump.

The dynamics of the SFB transmissions are assumed to follow the pseudo-rigid-body approximation with flexures approximated using pin-joints (Howell, 2001). Given this assumption, each SFB transmission has two inputs (forces generated by the actuators), and eight generalized coordinates. These include two independent coordinates (actuator tip deflections) and six dependent coordinates (a subset of flexure joint angles). The motion of the dependent coordinates is constrained by the parallel kinematics of the transmission. Thus, a complete model of the robot has eight inputs (actuator drive voltages), 38 generalized coordinates (76 states), and 24 position constraints. In addition, each transmission has a natural frequency experimentally determined to be near 100 Hz (Doshi et al., 2017). A more detailed description of the dynamic model for this robot is given by Doshi et al. (2018). The combination of a high-dimensional state space and high-frequency passive dynamics make trajectory optimization for HAMR particularly challenging.

7.1. Surface characterization

We experimentally verified the applicability of the Coulomb friction model to locomotion at HAMR’s scale. The coefficients of static friction (μ) between microrobot’s feet and three surfaces, PTFE (Teflon), card stock, and 1,200 grit sandpaper, were measured. Experiments were conducted using a single leg (Figure 8(a)) to closely replicate conditions during locomotion. Forces were measured by a six-axis force sensor (ATI Nano17 Titanium) at 100 Hz. Eight trials were run on each surface, and force traces for a representative trial are shown in Figure 8(b). The leg was manually lowered to pre-load the force sensor to 35–200% of the microrobot’s body weight (between 1 and 2 in Figure 8(b)). The swing degree of freedom is then actuated to generate a force in the x–y plane (3 in Figure 8(b)) until the leg begins to slip (4 in Figure 8(b)). The normal, F_n, and static frictional, F_f, forces were computed as

$$F_n = \Delta F_z \tag{36}$$

$$F_f = \sqrt{\Delta F_x^2 + \Delta F_y^2} \tag{37}$$

where ΔF_x, ΔF_y, and ΔF_z are the net forces between stages 1 and 4 in Figure 8(b) in the x, y, and z directions, respectively. The friction force increases linearly with the normal force as anticipated (Figure 8(c)). The mean and standard deviation for coefficients of friction averaged over the eight trials for Teflon, card stock, and 1,200 grit sandpaper are 0.29±0.03, 0.51±0.07, and 1.02±0.20, respectively. Lines corresponding to these average friction coefficients are shown in Figure 8(c).

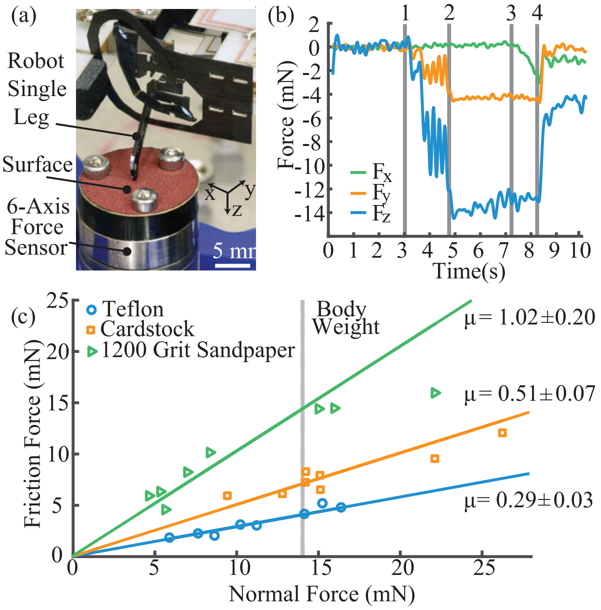

Fig. 8. (a) Labeled image of the friction-measurement experimental setup. (b) Representative force data: F_z is the normal force, and F_x and F_y are the tangential forces. (c) Raw data ($n = 8$) and best-fit lines corresponding to the estimated coefficients of static friction between the microrobot’s foot and the Teflon, card stock, and 1,200 grit sandpaper surfaces.

7.2. HAMR trajectory optimization

We performed a number of trajectory optimizations to search for periodic gaits that achieve forward velocities of approximately 10 mm per cycle near stride frequencies of 2, 10, and 30 Hz on three different surfaces: Teflon, card stock, and 1,200 grit sandpaper. A distance of 10 mm is nearly twice the robot’s step length, approaching the theoretical kinematic limit for a two-beat gait. In addition, the selected frequencies represent three different operational regimes for the microrobot as discussed by Goldberg et al. (2017b): quasi-static (2 Hz), near the vertical natural frequency (10 Hz) of the system, and near the roll natural frequency (30 Hz) of the system. Finally, these nine gaits cover a wide-range of ground contact conditions with coefficients of friction ranging from 0.29 to 1.02, demonstrating the versatility of the new algorithm.

The NLP presented in (34) was modified to search for periodic state and input trajectories by adding periodicity constraints on all position and velocity decision variables except the x-position of the body. We used the following cost function to encourage the robot to achieve its maximum theoretical stride length:

$$J = (x_N - x_g)^{\mathrm{T}} Q (x_N - x_g) + \sum_{i=2}^{N-1} \frac{c_1}{2} \Delta\dot{q}_i^{\mathrm{T}} \Delta\dot{q}_i + \frac{1}{2} \Delta u_i^{\mathrm{T}} \Delta u_i \tag{38}$$

where $\Delta\dot{q}_i = \dot{q}_i - \dot{q}_{i-1}$ and $\Delta u_i = u_i - u_{i-1}$ are the difference between neighboring generalized velocities and control inputs, respectively. Here Q is a diagonal matrix with $Q_{11} \in [10, 50]$ and the remaining diagonal entries equal to one, $c_1 \in [10, 50]$ scales the velocity difference penalty, and x_g is a goal state. To aid convergence and reduce susceptibility to local minima, the solver was initialized with an empirically generated trot gait that achieves roughly periodic motion for the card-stock friction coefficient. The goal state is set to $x_g = [10, x_p]^{\mathrm{T}}$, where $x_p \in \mathbb{R}^{75}$ is the periodic subset of the initial state.

A vertical jump trajectory of approximately one center-of-mass (COM) height was also optimized. The following cost function, which encourages the microrobot to jump to a specified height, was minimized:

$$J = (x_N - x_g)^{\mathrm{T}} Q (x_N - x_g) + \sum_{i=1}^{N-1} \frac{1}{2} u_i^{\mathrm{T}} R u_i \tag{39}$$

Here $x_g = [0_{2\times 1}, 24, 0_{73\times 1}]^{\mathrm{T}}$ is a goal state that specifies the desired apex height of the jump (slightly less than twice the COM height) with no body rotation or horizontal motion. The quadratic input cost penalizes swing actuator voltages as fore/aft forces do not contribute significantly to a vertical jump. To improve convergence time and avoid poor local optima, the optimization was initialized with a heuristically designed nominal jump trajectory.

As with the simulation examples, these ten trajectory optimization problems were implemented in MATLAB R2016a using the dynamics and control toolbox Drake. Due to the complexity of the HAMR model, these optimizations took several hours to converge.

7.3. Locomotion experiments

The actual performance of the microrobot when executing the planned trajectories from the previous section was evaluated in a controlled 20 cm × 20 cm motion-capture arena. Input signals were generated at 2.5 kHz using a MATLAB xPC environment (MathWorks, MATLAB R2015a), and were supplied to the microrobot through a 10-wire tether. Five motion capture cameras (Vicon T040) track the position and orientation of the robot body and the position of the feet at 500 Hz with a latency of 11 ms. In addition, eight piezoelectric encoders provide low-latency estimates of actuator tip velocities at 2.5 kHz (Jayaram et al., 2018). An estimator that combines foot-position and actuator-velocity measurements to generate low-latency estimates of the leg positions was developed. Finally, a PD controller was implemented to track the desired positions of the microrobot's four legs in the body-fixed frame. The details of the motion capture arena, estimator, and tracking controller are provided by Doshi et al. (2018).

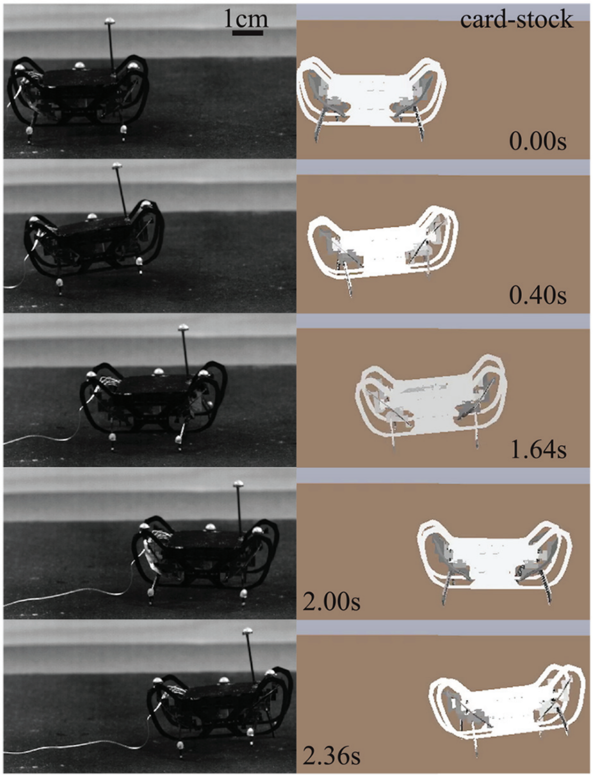

Fig. 9. Still frames of HAMR-VI (left) and a simulation of HAMR-VI (right) ambulating at 2 Hz on a card-stock surface.

7.4. Performance improvements

Control signals for periodic locomotion trajectories of HAMR were previously manually designed. This process involves tuning a series of parameters including input voltage, stride frequency, and relative phasing between the eight actuated degrees of freedom. Finding periodic control signals to produce a desired gait is often time consuming, requiring hundreds of experiments (Goldberg et al., 2017a,b). The new trajectory optimization method, in contrast, was able to generate open-loop control signals that offer improved performance without exhaustive experimentation.

The 2 Hz closed-loop experimental trajectories achieved an average velocity of 9.77 mm/cycle (Figure 10(a)), which is within 5% of the goal speed of 10 mm/cycle. These gaits also performed 26% better than the previous-best manually tuned gaits. In addition, the planned body pose closely matched that executed by the robot (Figure 9), demonstrating the accuracy of the trajectory optimization scheme. At 10 Hz, the closed-loop experimental trajectories achieved an average velocity of 8.98 mm/cycle, which is close to the desired velocity and 10% faster than the best manually

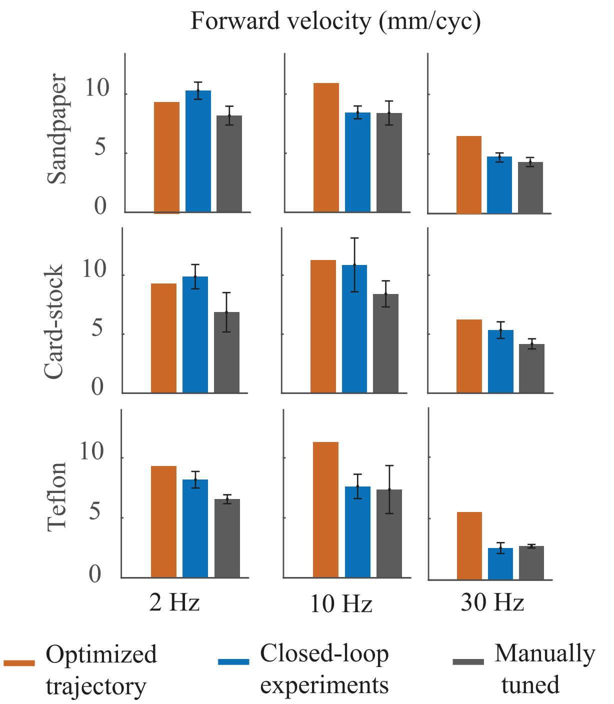

Fig. 10. Mean per-cycle forward velocity for the optimized (orange), closed-loop experimental (blue), and manually tuned (gray) trajectories. Error bars represent one standard deviation ($n = 15$).

tuned gaits. The card-stock gait at this frequency achieved the *fastest* per-cycle velocity recorded for this robot at 10.87 mm/cycle. Finally, the average velocity for the 30 Hz closed-loop gaits was slower at 4.24 mm/cycle. The closed-loop experiments on sandpaper and card stock, however, were still 20% faster than the best manually tuned gaits and within 20% of the predicted velocities from the optimizer. The robot also maintained the desired gait timing (front legs depicted in Figure 10(b)), and tracked the optimized leg trajectories in the body frame (front left leg depicted in Figure 10(c)) for all experiments except at 30 Hz on Teflon. Finally, we used this method to execute a vertical jump of 9.96 mm, which is approximately 78% of the robot's COM height (Figure 7). This was more than twice the jump height of 4.72 mm achieved using the heuristically designed jump trajectory that the optimizer was initialized with.

In summary, our model-based approach yielded improvements over previous experimental results (Goldberg et al., 2017a,b). Specifically, the average velocity of 9.21±1.31 mm/cycle achieved across the six gaits at 2 and 10 Hz is comparable with the highest previously measured experimental velocity of 9.5 mm/cycle achieved using careful hand tuning on a card-stock surface (Goldberg et al., 2017a). Even the three slower 30 Hz gaits move on average 30% faster than previously recorded trots at similar frequencies on a card-stock surface. In addition, the robot was able to achieve a new highest velocity of 10.87 mm/cycle, and demonstrate the first controlled vertical jump of 9.96 mm (78% of COM height). Importantly, these performance improvements were achieved without exhaustive experimentation: tens of experiments were conducted as opposed to hundreds.

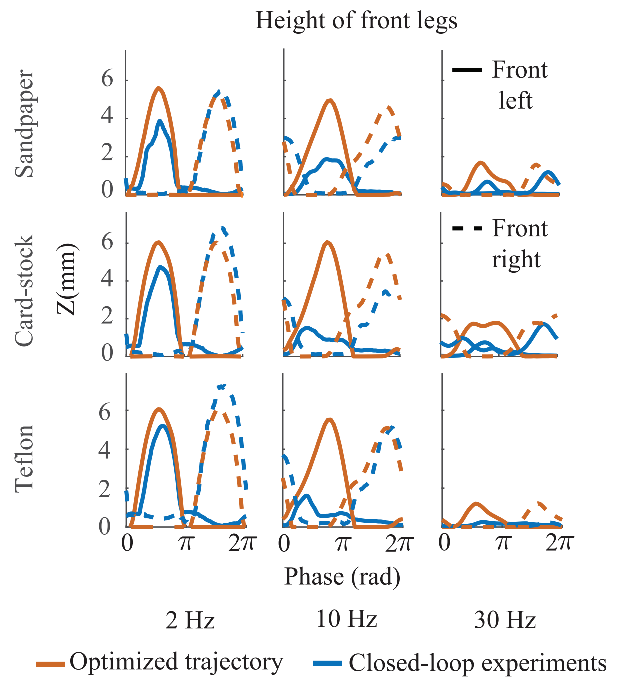

Fig. 11. Optimized (orange) and mean closed-loop experimental (blue) leg height for the front left (solid) and front right (dashed) leg.

7.5. Quality of optimized trajectories

We evaluated the quality of the periodic trajectories by measuring the normalized average slip, $\bar{s}$, commonly defined as (Jayaram and Full, 2016; Ridgel et al., 2003)

$$\bar{s} = \frac{1}{4\int_{t_0}^{t_f} v_x(t)dt} \sum_{i=1}^{4} \int_{\xi} |v_x^i(t)|dt \tag{40}$$

Here, v_x^i is the x-velocity of the ith leg and v_x is the x-velocity of the COM, both in the world-fixed frame as measured by the motion capture system. The time interval of interest is bounded by t_0 and t_f, and ξ is the set of times for which $v_x^i < 0$. Normalized slip is the total distance a single leg travels backwards in the world frame divided by the forward distance traveled by the body. We present an average value for all four legs. Higher values of $\bar{s}$ indicate increased backwards motion of the legs, decreased propulsion, and reduced performance. The average value of $\bar{s}$ was 0.10±0.06 ($n = 9$) for the optimized trajectories, which is expected because we demand high performance from the robot. The closed-loop experimental trajectories slipped slightly more, with an average $\bar{s}$ of 0.24±0.14 ($n = 9$). This is similar to the average slip of 0.23±0.11 ($n = 9$)

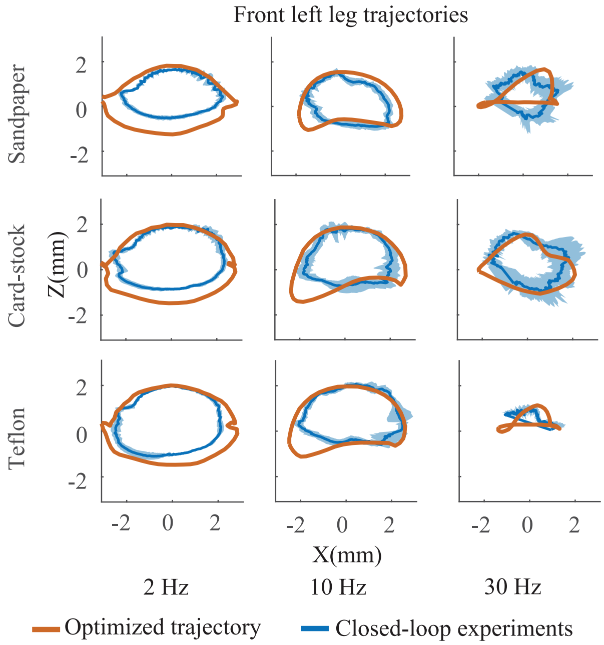

Fig. 12. Optimized (orange) and mean closed-loop experimental (blue) periodic foot trajectories for the front left leg. The blue shaded region represents one standard deviation ($n = 15$).

across the manually tuned trajectories, and is one of the factors that could have resulted in decreased performance when compared with the optimized trajectories. In addition, the optimizer also found an intuitive jumping trajectory where all four legs first build spring potential energy, and then simultaneously push into the ground.

8. Discussion and conclusions

We have presented a new family of variational time-stepping algorithms that generalize previous methods to higher orders of integration accuracy. We used a midpoint discretization to derive a second-order method and incorporated it into a direct trajectory optimization algorithm that solves for contact forces along with state and input trajectories. We also demonstrated the algorithm's ability to generate walking trajectories for both simulated underactuated robots and a physical quadrupedal microrobot with complex dynamics. Hardware experiments on this microrobot showed significant improvements over previous hand-tuned trajectories.

Numerical tests suggest that our second-order contact-implicit method offers improved accuracy over previous first-order algorithms, allowing smaller NLPs to be solved while maintaining reasonable solution accuracy. Compared with existing hybrid trajectory optimization methods, our technique can offer comparable solution accuracy without requiring pre-specification of a contact mode sequence. In particular, if the slack variables in (34) converge to zero, the solution from our method should correspond to a hybrid method supplied with the same mode sequence. However, the size of the optimization problem that must be solved in the contact-implicit formulation is larger than in a hybrid method, potentially leading to significantly longer solution times.

Several directions remain for future work. For example, a more extensive numerical comparison including third- and fourth-order time-stepping methods would allow us to better understand the trade-off between accuracy and computational cost in the context of motion planning. Furthermore, our current MATLAB implementation requires several minutes to compute the Spring Flamingo and LittleDog plans and several hours to compute the HAMR-VI plans described in the previous section. Significant speed improvements could be made with a careful C++ implementation that exploits the sparsity structure of the problem. Lastly, there may also be advantages to combining our contact-implicit approach with existing hybrid trajectory optimization schemes such as that of Posa et al. (2016) by warm-starting the hybrid method with the contact mode sequence and trajectory discovered by our method. This combined approach could ultimately offer faster convergence to more accurate solutions.

Acknowledgments

The authors would like to thank Professor Michael Posa, Professor Inna Sharf, Dr. Kaushik Jayaram, Dr. Benjamin Goldberg, and the members of the Harvard Agile Robotics and Microrobotics Labs for their helpful comments and discussions.

Funding

This work was supported by a Draper Internal Research and Development grant, a Google Faculty Research Award, the Wyss Institute for Biologically Inspired Engineering, the National Defense Science and Engineering Graduate Fellowship and, the National Science Foundation (grant number IIS-1657186). In addition, the prototypes were enabled by equipment supported by the ARO DURIP program (award #W911NF-13-1-0311). Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

ORCID iDs

Zachary Manchester https://orcid.org/0000-0002-3071-7091
Neel Doshi https://orcid.org/0000-0001-7011-0836

References

Anitescu M (2005) On using the elastic mode in nonlinear programming approaches to mathematical programs with complementarity constraints. *SIAM Journal on Optimization* 15(4): 1203–1236.

Anitescu M and Potra FA (1997) Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics* 14(3): 231–247.

Betts JT (1998) Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics* 21(2): 193–207.

Boyd S and Vandenberghe L (2004) *Convex Optimization*. Cambridge: Cambridge University Press.

Buchli J, Pratt J, Roy N, et al. (2010) The LittleDog robot. *The International Journal of Robotics Research* 30(2): 145–149.

Buss M, Hardt M, Kiener J, et al. (2003) Towards an autonomous, humanoid, and dynamically walking robot: modeling, optimal trajectory planning, hardware architecture, and experiments. In: *Proceedings of the 3rd International Conference on Humanoid Robotics*, pp. 2491–2496.

Doshi N, Goldberg B, Sahai R, Jafferis N, Aukes D and Wood RJ (2015) Model driven design for flexure-based microrobots. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4119–4126.

Doshi N, Jayaram K, Goldberg B, Manchester Z, Wood RJ and Kuindersma S (2018) Contact-implicit optimization of locomotion trajectories for a quadrupedal microrobot. In: *Robotics: Science and Systems (RSS)*.

Doshi N, Jayaram K, Goldberg B and Wood RJ (2017) Phase control for a legged microrobot operating at resonance. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5969–5975.

Fletcher R and Leyffer S (2004) Solving mathematical programs with complementarity constraints as nonlinear programs. *Optimization Methods and Software* 19(1): 15–40.

Gill PE, Murray W and Saunders MA (2005) SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review* 47(1): 99–131.

Goldberg B, Doshi N, Jayaram K, Koh JS and Wood RJ (2017a) A high speed motion capture method and performance metrics for studying gaits on an insect-scale legged robot. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3964–3970.

Goldberg B, Doshi N, Jayaram K and Wood RJ (2017b) Gait studies for a quadrupedal microrobot reveal contrasting running templates in two frequency regimes. *Bioinspiration and Biomimetics* 12(4): 046005.

Goldstein H, Poole C and Safko J (2001) *Classical Mechanics*. 3rd Ed. San Francisco, CA: Addison Wesley.

Howell LL (2001) *Compliant Mechanisms*. New York: John Wiley & Sons, Inc.

Jafferis NT, Smith MJ and Wood RJ (2015) Design and manufacturing rules for maximizing the performance of polycrystalline piezoelectric bending actuators. *Smart Materials and Structures* 24(6): 065023.

Jayaram K and Full RJ (2016) Cockroaches traverse crevices, crawl rapidly in confined spaces, and inspire a soft, legged robot. *Proceedings of the National Academy of Sciences* 113(8): E950–E957.

Jayaram K, Jafferis NT, Doshi N, Goldberg B and Wood RJ (2018) Concomitant sensing and actuation for piezoelectric microrobots. *Smart Materials and Structures*, in press.

Junge O, Marsden JE and Ober-Blöbaum S (2005) Discrete mechanics and optimal control. *IFAC Proceedings Volumes* 38(1): 538–543.

Manchester Z and Kuindersma S (2017) Variational contact-implicit trajectory optimization. In: *Proceedings of the International Symposium on Robotics Research (ISRR)*, Puerto Varas, Chile.

Marsden J and Ratiu T (1999) *Introduction to Mechanics and Symmetry* (*Texts in Applied Mathematics*). 2nd Ed. New York: Springer.

Marsden JE and West M (2001) Discrete mechanics and variational integrators. *Acta Numerica* 10: 357–514.

Mayne DQ (1966) A second-order gradient method of optimizing non-linear discrete time systems. *International Journal of Control* 3: 8595.

Mombaur KD (2009) Using optimization to create self-stable human-like running. *Robotica* 27(3): 321–330.

Mordatch I, Todorov E and Popović Z (2012) Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics* 31(4): 43–43:8.

Moreau J (1973) On unilateral constraints, friction and plasticity. In: *New Variational Techniques in Mathematical Physics*. New York: Springer, pp. 172–322.

Neunert M, Farshidian F and Buchli J (2016) Efficient whole-body trajectory optimization using contact constraint relaxation. In: *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 43–48.

Neunert M, Farshidian F, Winkler AW and Buchli J (2017) Trajectory optimization through contacts and automatic gait discovery for quadrupeds. *IEEE Robotics and Automation Letters* 2(3): 1502–1509.

Nocedal J and Wright SJ (2006) *Numerical Optimization*. 2nd Ed. New York: Springer.

Ober-Blobaum S and Saake N (2015) Construction and analysis of higher order Galerkin variational integrators. *Advances in Computational Mathematics* 41: 955–986.

Posa M, Cantu C and Tedrake R (2014) A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research* 33(1): 69–81.

Posa M, Kuindersma S and Tedrake R (2016) Optimization and stabilization of trajectories for constrained dynamical systems. In: *Proceedings of the International Conference on Robotics and Automation (ICRA)*. Stockholm, Sweden: IEEE, pp. 1366–1373.

Pratt J and Pratt G (1998) Intuitive control of a planar bipedal walking robot. In: *Proceedings 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, Vol. 3, pp. 2014–2021.

Remy CD (2011) *Optimal Exploitation of Natural Dynamics in Legged Locomotion*. PhD Thesis, ETH ZURICH.

Ridgel AL, Ritzmann RE and Schaefer PL (2003) Effects of aging on behavior and leg kinematics during locomotion in two species of cockroach. *Journal of Experimental Biology* 206(24): 4453–4465.

Schultz G and Mombaur K (2010) Modeling and optimal control of human-like running. *IEEE/ASME Transactions on Mechatronics* 15(5): 783–792.

Stewart DE (2000) Rigid-body dynamics with friction and impact. *SIAM Review* 42(1): 3–39.

Stewart DE and Trinkle JC (1996) An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and Coulomb friction. *International Journal for Numerical Methods in Engineering* 39(15): 2673–2691.

Tassa Y, Erez T and Todorov E (2012) Synthesis and stabilization of complex behaviors through online trajectory optimization. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Tedrake R and the Drake Development Team (2016) Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems. Available at: https://wiki.qut.edu.au/download/attachments/173981546/drake.pdf?version=1&modificationDate=1403592695000&api=v2

Todorov E (2011) A convex, smooth and invertible contact model for trajectory optimization. In: *Proceedings of the International Conference on Robotics and Automation (ICRA)*.

Vanderbei RJ and Yurttan H (1998) Using LOQO to solve second-order cone programming problems. Technical Report SOR-98-9, Princeton University, Princeton, NJ.

Whitney JP, Sreetharan PS, Ma KY and Wood RJ (2011) Pop-up book MEMS. *Journal of Micromechanics and Microengineering* 21(11): 115021.

Xi W and Remy CD (2014) Optimal gaits and motions for legged robots. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.