

This paper has been accepted for publication in IEEE Robotics and Automation Letters.

DOI: 10.1109/LRA.2019.2900840

IEEE Explore: <https://ieeexplore.ieee.org/document/8648229/>

Please cite the paper as:

Amir Patel, Stacey Shield, Saif Kazi, Aaron M. Johnson, and Lorenz T. Biegler, “Contact-Implicit Trajectory Optimization Using Orthogonal Collocation,” in *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2242–2249, April 2019.

©2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. DOI: 10.1109/LRA.2019.2900840

Contact-Implicit Trajectory Optimization using Orthogonal Collocation

Amir Patel¹, Stacey Shield¹, Saif Kazi², Aaron M. Johnson³, and Lorenz T. Biegler²

Abstract—In this paper we propose a method to improve the accuracy of trajectory optimization for dynamic robots with intermittent contact by using orthogonal collocation. Until recently, most trajectory optimization methods for systems with contacts employ mode-scheduling, which requires an a priori knowledge of the contact order and thus cannot produce complex or non-intuitive behaviors. Contact-implicit trajectory optimization methods offer a solution to this by allowing the optimization to make or break contacts as needed, but thus far have suffered from poor accuracy. Here, we combine methods from direct collocation using higher order orthogonal polynomials with contact-implicit optimization to generate trajectories with significantly improved accuracy. The key insight is to increase the order of the polynomial representation while maintaining the assumption that impact occurs over the duration of one finite element.

Index Terms—Motion and Path Planning, Contact Modeling

I. INTRODUCTION

CONTACT is ubiquitous in nature. Animals utilize contact with their environment to enable dexterous manipulation of objects or agile locomotion. In order for robots to one day perform similarly complex maneuvers in challenging environments, contacts must be adequately considered (and exploited) during motion planning. However, achieving this is still challenging as impulsive contact represents a significant numerical challenge for current methods.

One recent approach to solving this problem, introduced by Posa [1] and Mordatch [2], is *contact-implicit optimization* (also called *contact-invariant optimization* or *through-contact optimization*). This method encodes both the contact force and the body state as part of an optimization problem, with the contact consistency enforced by algebraic constraints. Unlike hybrid systems models, which use a fixed contact sequence defined a priori (or with an outer loop optimization) for the full system [3–7] or per-leg [8], or Mixed Integer Programming (MIP) [9, 10], which encodes the contact mode in an integer variable, in contact-implicit optimization there is no variable specifying the contact mode at each time, and the contact

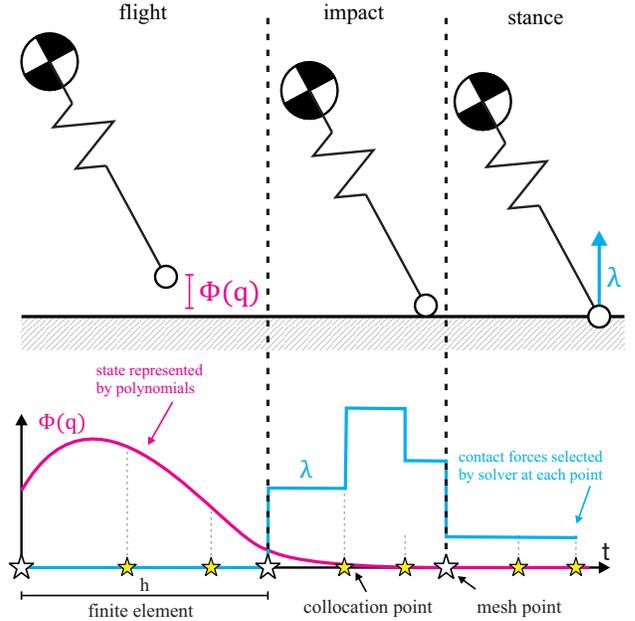


Fig. 1. The concept of contact-implicit optimization using orthogonal collocation is shown. The state trajectory is expressed as a series of high-order polynomials on finite elements. Contact mode changes are enforced to only occur at the edges (mesh points) of the finite elements, ensuring smoothness and accuracy of the transcription.

is determined implicitly by the optimization. These methods are very effective for systems with many possible contacts or an unknown optimal contact sequence, and have enabled, e.g., discoveries of novel legged gaits [11, 12] and balancing trajectories [13].

However, most implementations only employ a first-order (Euler) integration method [1, 2, 13] or a pseudo-trapezoidal method [11, 14], and are based on time-stepping simulation [15]. The disadvantage of first-order integration is that these methods require a large number of steps, N , for sufficient accuracy (they have $O(1/N)$ accuracy), which limits their application to short time-horizon motions. These methods have been combined with variational integrators to produce a contact-implicit method with second-order accuracy [16], as well as versions that use B-Splines [17] or Hermite-Simpson polynomials [18]. Previous works have noted that time-stepping methods can never exceed first-order discretization due to the fact that discontinuities could exist within the state trajectories [14, 19–21].

In this paper, we show that higher-order methods can indeed be employed and that doing so increases the accuracy of contact-implicit trajectory optimization. Our method uses

Manuscript received: September, 10, 2018; Revised December, 12, 2018; Accepted January, 30, 2019.

This paper was recommended for publication by Editor Nancy Amato upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the National Research Foundation of South Africa (Grant: 99380) and the Oppenheimer Memorial Trust.

¹Amir Patel and Stacey Shield are with the Department of Electrical Engineering, University of Cape Town, South Africa. a.patel@uct.ac.za

²Saif Kazi and Lorenz T. Biegler are with the Department of Chemical Engineering, Carnegie Mellon University, USA. bieglert@cmu.edu

³Aaron M. Johnson is with the Department of Mechanical Engineering, Carnegie Mellon University, USA. amj1@cmu.edu

Digital Object Identifier (DOI): 10.1109/LRA.2019.2900840

higher-order orthogonal collocation (i.e. it solves the dynamics at collocation points based on K -th degree orthogonal polynomials) [22, 23] to provide a smoother representation of the state and velocity at every point on the interior of a finite element (for $O((1/N)^{2K-1})$ accuracy).

It is this more accurate smooth representation that [14, 19–21] point to as the problem with higher-order methods, because a smooth representation would require an event-finding algorithm (i.e. collision detection) to isolate the non-smooth points. The presented method resolves this problem by approximating an event-driven simulation scheme (instead of time-stepping) within the context of a contact-implicit optimization problem. That is, the presented method isolates the event times that the algebraic constraints are activated or deactivated and can apply an impulsive transition between contact modes. The key to this working is that we constrain the events (contact mode switches) to occur only at the mesh points (edges of the finite elements) as shown in Fig. 1. So long as the optimizer is given sufficient freedom to adjust the times that these mesh points represent, it implicitly solves the event-finding problem as part of the optimization. This eliminates the need for an additional collision detection algorithm. To make the problem formulation more tractable, we also present a relaxation that still maintains several key physical assumptions (especially impulses acting over the duration of one finite element) from the first-order implementations. We show that this relaxation approximates the exact formulation in the limit as time-steps are allowed to go to zero.

The paper is organized as follows: Sec. II describes the method, starting with a background on collocation for dynamic (but smooth) trajectory optimization (Sec. II-A) and then contact-implicit trajectory optimization (Sec. II-B). In Sec. II-C, the application of orthogonal collocation to contact-implicit optimization problems is presented, with further implementation considerations presented in Sec. II-D. Then, Sec. III presents three case studies which utilize the method, ranging from a simple point particle to a bipedal robot. Lastly, Sec. IV concludes the paper with a discussion of the implications of the results and proposed future work.

II. METHOD

This paper considers trajectory optimization, where the dynamics and control inputs are transcribed into a nonlinear program (NLP) that can be solved with numerical optimization algorithms. Trajectory optimization can be formulated either by *direct* or *indirect* methods. Here, we focus on direct methods as indirect methods have known disadvantages [3]. Comprehensive tutorials on trajectory optimization can be found in [3, 23].

A. Direct Collocation

Direct collocation formulates the trajectory optimization problem as an NLP without the need for forward integration (as in shooting methods) [24]. This is done by discretizing the trajectories (state and control) into N time periods (finite elements) using polynomials. In our particular case the trajectories are represented using a Runge-Kutta basis with K -collocation points. The advantage of this representation, when

compared to others such as B-splines [17], is that most of the polynomial coefficients have the same variable bounds as the profiles themselves as well as considerably better numerical accuracy [23, Ch. 8].

For example, consider the state variable z :

$$\frac{dz}{dt} = f(z(t), t), \quad z(0) = z_0. \quad (1)$$

For time t in finite element i , this yields the following Runge-Kutta basis representation of the state variable:

$$z(t) = z_{i,0} + h_i \sum_{j=1}^K \Omega_j(\tau) \dot{z}_{ij}, \quad t \in [t_i, t_{i+1}], \quad (2)$$

where $z_{i,0}$ is a coefficient that represents the state variable at the beginning of element i , \dot{z}_{ij} represents $\frac{dz(t_{ij})}{d\tau}$, h_i is the length of the finite element, τ the relative time within that element, and $\Omega_j(\tau)$ is a polynomial of order K , satisfying:

$$\Omega_j(\tau) = \int_0^\tau \bar{l}(\tau') d\tau', \quad \tau \in [0, 1], \quad (3)$$

where $\bar{l}(\tau') = \prod_{k=1, \neq j}^K \frac{(\tau' - \tau'_k)}{(\tau'_j - \tau'_k)}$.

Here, we employ K -point Radau collocation (a Gauss-Jacobi polynomial) to solve the differential equation at selected points in time. Radau collocation has many attractive features (stability, stiff decay) as well as having equivalent accuracy to Implicit Runge Kutta (IRK) integration, $O(h^{2K-1})$ [23]. Using this, the state variable at each collocation point k of finite element i is represented as:

$$z_{i,k} = z_{i,0} + h_i \sum_{j=1}^K \Omega_j(\tau_k) \dot{z}_{ij}, \quad k \in \{1, \dots, K\}, \quad (4)$$

with Ω derived using (3) and τ_k the relative time of collocation point k . For a given K , values for the weightings Ω and time divisions τ can be found in [23, Ch. 8]. Continuity at the finite element boundaries is enforced by:

$$z_{i,0} = z_{i-1,K}, \quad i \in \{2, \dots, N\}. \quad (5)$$

B. Contact-Implicit Trajectory Optimization

For contact-implicit trajectory optimization, the dynamics (1) can be modeled as a rigid multi-body system using Euler-Lagrange mechanics with generalized coordinates \mathbf{q} and often expressed in the form:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{G} = \mathbf{B}\mathbf{u} + \mathbf{J}^T\boldsymbol{\lambda}, \quad (6)$$

where \mathbf{M} represents the mass matrix, \mathbf{C} the Coriolis and centrifugal matrix, \mathbf{G} the gravitational force, \mathbf{B} the input mapping, \mathbf{u} the generalized input, \mathbf{J} the contact Jacobian, and $\boldsymbol{\lambda}$ the contact forces. Multi-body systems with contact possess hybrid dynamics, where $\boldsymbol{\lambda}$ only acts in specific configurations of the state space (i.e. when in contact). When switching between contact conditions, Newtonian plastic impact says that an impulse Λ at the contact location leads to a discontinuity

in velocity from $\dot{\mathbf{q}}^-$ (pre-impact) to $\dot{\mathbf{q}}^+$ (post-impact), defined by¹:

$$\mathbf{M}(\dot{\mathbf{q}}^+ - \dot{\mathbf{q}}^-) = \mathbf{J}^T \Lambda, \quad \mathbf{J}\dot{\mathbf{q}}^+ = 0 \quad (7)$$

which may be derived by taking the limit of an impact event as the time duration goes to zero in (6).

Prior contact-implicit methods of trajectory optimization for these systems [1, 2] are based off the time-stepping simulation method [15] which discretizes time and considers the combination of both forces λ_i and impulses Λ_i integrated over time-step i in a combined λ_i . The dynamics are encoded using direct transcription using 1-point collocation, i.e. a first order approximation defined entirely by the value of the states and contact forces at each mesh point (the start of each finite element). By including λ as part of the decision variables, the optimization implicitly chooses the sequence of contacts, represented as time-steps i where $\lambda_i > 0$. The contact force requires the complementarity constraint:

$$\lambda_i^T \phi(\mathbf{q}_{i+1}) = 0, \quad \phi(\mathbf{q}_i) \geq 0, \quad \lambda_i \geq 0 \quad (8)$$

where $\phi(\mathbf{q})$ represents the non-penetration constraint between rigid bodies (and $\mathbf{J} = \frac{\partial \phi}{\partial \mathbf{q}}$). Note that this formulation is not meant for systems with a large number of bodies or complex surfaces (e.g. billiards or walking on gravel) and assumes a finite number of contact constraints. The choice of indices is important to ensure that a contact force or impulse over a time-step i enforces the equality $\phi_{i+1} = 0$ at the end of that time-step. Additional constraints enforce the friction cone² [1]:

$$\lambda_{y,i} \geq 0, \quad \lambda_{x,i}^+ \geq 0, \quad \lambda_{x,i}^- \geq 0, \quad (9)$$

$$\mu \lambda_{y,i} - \lambda_{x,i}^+ - \lambda_{x,i}^- \geq 0, \quad (10)$$

$$(\mu \lambda_{y,i} - \lambda_{x,i}^+ - \lambda_{x,i}^-)^T \gamma_i = 0, \quad (11)$$

with $\lambda = [\lambda_{x,i}^+ - \lambda_{x,i}^-, \lambda_{y,i}]^T$ where x is the direction tangent to the contact surface and y is the direction normal to it, while γ_i is the magnitude of the relative tangential velocity at the point of contact. Additionally, if the contact point is sliding, it is constrained to do so with a frictional force along the edge of the friction cone:

$$\gamma_i + \psi(q_i, \dot{q}_i) \geq 0 \quad (12)$$

$$\gamma_i - \psi(q_i, \dot{q}_i) \geq 0 \quad (13)$$

$$\lambda_{x,i}^{+T} (\gamma_i + \psi(q_i, \dot{q}_i)) = 0 \quad (14)$$

$$\lambda_{x,i}^{-T} (\gamma_i - \psi(q_i, \dot{q}_i)) = 0 \quad (15)$$

with $\psi(q_i, \dot{q}_i)$ the relative tangential velocity at the contact.

The constraints (8), (11), (14), (15) transform the NLP into a Mathematical Program with Equality Constraints (MPEC) which is notoriously difficult to solve. The two main methods for making the MPEC problem more tractable are the ε -relaxation method and the penalty method [26–28]. In the

ε -relaxation method, the complementarity constraints are reformulated as a set of inequality constraints, relaxed by a parameter $\varepsilon > 0$:

$$\alpha^T \beta = 0 \Rightarrow \alpha^T \beta \leq \varepsilon, \quad (16)$$

where α and β are positive slack variables. The MPEC is then solved as series relaxed problems decreasing ε to a user-defined accuracy. In the penalty method [29], the complementarity constraint is removed and its l_1 norm is included in the objective:

$$\min_z g(z) + \rho \alpha^T \beta. \quad (17)$$

This allows the problem to appear more feasible to the NLP, but requires ρ to be greater than some critical value ρ_c in order to exactly satisfy the complementarity at the solution.

In the examples described in Section III, we find that the best MPEC solution strategy is problem dependent.

C. Our Approach

In this paper, we use direct collocation with higher-order orthogonal polynomials to represent the state \mathbf{q} and velocity $\dot{\mathbf{q}}$ of a contact-implicit optimization, while enforcing the dynamics at both the mesh points and the collocation points. Prior first-order methods evaluate the dynamics at each mesh point and use a linear interpolation in between (Fig. 2). These first-order methods do not determine where within a finite element an impact occurs, because the effect of an impulse Λ is spread out over the entire element. However, only having a linear interpolation limits their accuracy both by not locating the point of impact as well as not following other nonlinear changes in the trajectory.

Here, with a higher-order representation of the system within a finite element, contact changes are constrained to only occur at the mesh points. If this is not done, as in [17, 18], complementarity is not guaranteed within the finite element (e.g. the foot could leave or strike the ground at a collocation point). Using a higher-order representation with contact changes at the mesh points increases the accuracy of the trajectory both by isolating the time of impacts and by tracking the continuous dynamics more closely.

By isolating the event times, the optimization can be considered as approximating an event-driven simulation scheme instead of a time-stepping scheme. In event-driven simulation, the continuous dynamics are integrated with a numerical ODE (ordinary differential equation) or DAE (differential-algebraic equation) integration algorithm (e.g. `ode45`) and stopped when an event is detected to handle the impulsive change in contact conditions. A time-stepping scheme combines the effect of continuous dynamics over a fixed time-step with the impulsive contact changes in a single step, performing a first-order integration of the system as an MDI (measure differential inclusion). These are typically considered separate classes of numerical algorithms (e.g. [30, Sec. 6.3]), but in this setting lying along the same continuum.

Once the events are constrained to the mesh points, the question then becomes how to handle the discontinuous dynamics of impact, (7). This was not an issue in the first-order methods since they are already non-smooth everywhere.

¹In closed form, these constraints define $\Lambda = -(\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}\mathbf{J}\dot{\mathbf{q}}^-$, as in e.g. [25, Eqn. 25], however in this optimization context it is more convenient to leave the definition of the impulse Λ implicit. While other models of impact dynamics could be used, we believe that plastic impact is the most appropriate model for robot dynamics for the reasons given in [25, A8].

²For clarity, we've only included described the 2D case, but these can easily be extended to 3D [16].

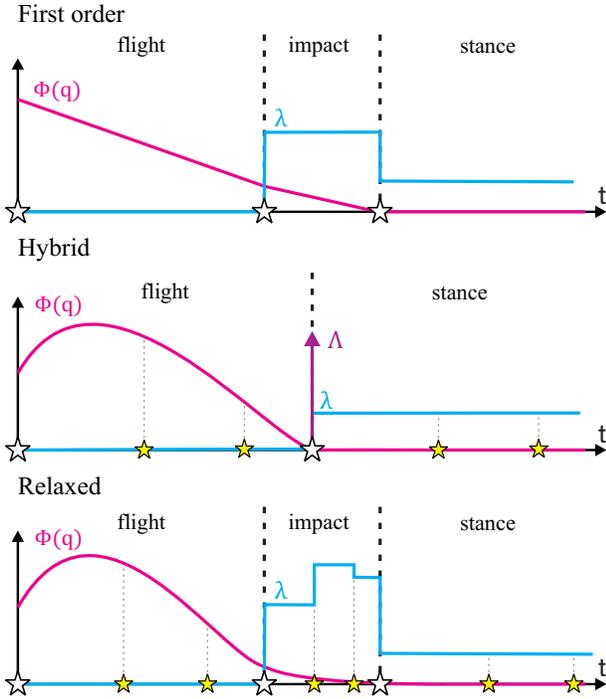


Fig. 2. Comparison of different approaches to handling impact. In first order methods, the state is piecewise-linear and the effect of contact forces and impulses are combined over a finite element. In the hybrid-system formulation, the state is a smooth polynomial within a finite element and the optimization solves for the contact forces, impulses, and times. In the relaxed formulation, the smooth state representation is maintained everywhere but the contact impulse is again combined with contact force over a finite element. The impact element is exaggerated in length for clarity.

Here we present two possible solutions: the full hybrid-system formulation and a more tractable relaxed approximation.

a) *Hybrid-system Formulation*: The hybrid-system dynamics of (7) can be explicitly included in the optimization by replacing the velocity continuity equations (5). Contact impulses Λ_i at each mesh point i are added to the set of decision variables (Fig. 2), just as contact forces λ_i are in any contact-implicit scheme, with the additional constraints:

$$\Lambda_i^T \phi(\mathbf{q}_{i+1}) = 0, \quad \phi(\mathbf{q}_i) \geq 0, \quad \Lambda_i \geq 0 \quad (18)$$

as well as constraints analogous to (9)–(11) in the case of frictional impact.

b) *Relaxed Formulation*: However, this hybrid formulation introduces even more complementarity conditions and thus far has only resulted in a solvable optimization problem when initialized very close to the correct solution. Therefore, we relax the hard-impact constraint by spreading the impulse out over the duration of a finite element (Fig. 2). This may at first seem to violate the rigid-body model of physics, as the impulse start to act before the object reaches the contact. However, this is the same relaxation that the first-order methods use implicitly [1, 2], the only difference is that the higher-order version approximates the intermediate trajectory smoothly while the first-order case does not. The other contact constraints still hold – the impulse may act just before contact is made only if the bodies do in fact make contact at the end of that finite element.

The impulse (encoded as part of the contact force) is only allowed to be applied for the duration of one finite element. That time, h_i , is a decision variable. In the limit, if we let h_i become very small, the equations of motion converge to exactly the plastic impact law of (7), as seen in Sec. III-A and Fig. 3. In practice, the optimization algorithm naturally chooses small h_i with comparatively large contact forces as it uses this freedom to approximate the rigid-body impact.

Using the complementarity formulation of (8), requiring contact over finite element $i + 1$ to enable contact forces over finite element i , also produces a challenging constraint at liftoff. Just as the contact force could act before touchdown, it must also cease before liftoff. Smooth (nonimpulsive) liftoff occurs when the contact force goes to zero anyway, so this constraint is not as tricky as the touchdown constraint. However, the optimizer must have the freedom to either use sufficient control effort to maintain zero contact force or reduce the time duration of the liftoff event to a small value of h_i . This also implies that contacts must persist for a minimum dwell time of at least one finite element, precluding exact solutions to simultaneous but sequential transitions (e.g. [25, Thm. 8]). An example showing this limitation is given in Sec. III-A.

D. Implementation Details

A previous application of orthogonal collocation to MPEC optimizations by Baumrucker & Biegler [29] enforces mode changes at the mesh points by complementing one variable with the L_1 norm of the other within the finite element. A new slack variable is introduced:

$$\alpha'_i = \sum_{j=0}^K \alpha_{ij}, \quad (19)$$

and then the complementarity is expressed as:

$$\alpha'_i \beta_{ij} = 0. \quad (20)$$

This solves the complementarity constraint at each collocation point and ensures that the mode is constant within the finite element. This formulation ensures accuracy but results in $N \times K$ complementarity equations.

To apply this to contact-implicit optimization, we propose an additional set of slack variables:

$$\beta'_i = \sum_{j=0}^K \beta_{ij}, \quad (21)$$

with the following reformulated complementarity:

$$\alpha'_i \beta'_i = 0, \quad (22)$$

which increases the problem size, but results in the complementarity constraints only being evaluated once at each mesh point, while ensuring contact mode is fixed within the finite element. This formulation can readily be applied to the complementarity equations (8), (11), (14) and (15).

An additional change that must be introduced when increasing the number of collocation points is that the control input, \mathbf{u} , must be constrained within a finite element. While the control variables could be represented using Lagrange polynomials, with discontinuities at the mesh points, in practise this leads

to an artificial oscillation in control signal over the course of a finite element which leads to slow convergence (see [23] for discussion). Here, we employ a piecewise-constant control within the finite element as this mitigates the problem of singular arcs in the control problem (though other more refined control laws may be used instead). An example of why this is necessary is given in Sec. III-C and Fig. 7.

In summary, the optimization problem has the following decision variables, indexed by system state h , finite element i , collocation point j , contact point k , control input l , and direction $m \in \{x, y\}$:

$q_{h,i,j}$	System state
$\dot{q}_{h,i,j}$	System velocity
$\ddot{q}_{h,i,j}$	System acceleration
$\lambda_{y,i,j,k}$	Normal contact force
$\lambda_{x,i,j,k}^+$	Tangential positive contact force
$\lambda_{x,i,j,k}^-$	Tangential negative contact force
$u_{i,l}$	Control input
h_i	Time duration
$\alpha_{m,i,j}, \alpha'_{m,i}$	slack variables for $\phi_{i,j,k}$ or $\gamma_{i,j,k}$
$\beta_{m,i,j}, \beta'_{m,i}$	slack variables for $\lambda_{m,i,k}$

For H system states, N finite elements, K point collocation, and C contact points, and U control inputs, this results in $N(K(3H + 3C) + U + 1 + 8C(K + 1))$ decision variables (for planar friction).

The optimization problem has the following constraints:

- (6) Acceleration dynamics
- (4), (5) Collocation constraints for q, \dot{q}
- (8) Normal complementarity
- (9) – (15) Frictional complementarity
- (19), (21) Slack variable definitions and constraints

for a total of $N(3KH + C(20 + 8K))$ constraints, in addition to problem-specific constraints such as initial and final conditions, bounds on variables such as h_i , or input constraints. The objective function, g , is also problem-specific, with minimum-time, minimum-effort, and similar functions commonly used. This results in a large but sparse optimization problem which is particularly suited for sparse nonlinear solvers such as IPOPT [31], CONOPT [32], or SNOPT [33]. For the results in this paper, the optimization problems were written in GAMS [34] and solved with either IPOPT or CONOPT. This code is available online:

<https://github.com/UCTMechatronics/orthogonal-collocation-with-contacts>

III. RESULTS

We implement the contact-implicit trajectory optimization with orthogonal collocation method on three example problems, with a focus on testing the hypothesis that it provides better accuracy. We also show the implications of some of the formulation decisions discussed in Sec. II-C and II-D.

A. Ball Hitting Ceiling

In this example, we simulate the trajectory of a ball (point mass) colliding with a ceiling. The ball has an initial upward velocity and is acted on by gravity and a contact force when colliding with the ceiling. This example was chosen to

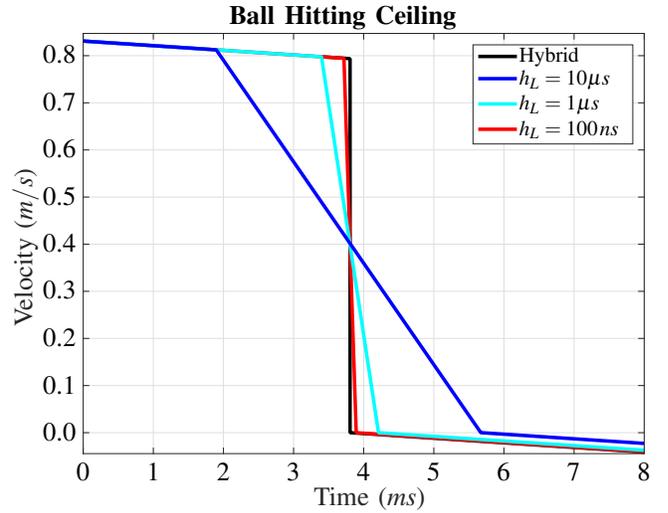


Fig. 3. Velocity at the point of impact is depicted. As the time-step lower bound (h_L) is decreased, the solution approaches the true, discontinuous solution.

demonstrate the ability to successfully capture impulse-like behavior as the lower bound on finite element time, $h_L \leq h_i$, is reduced. The system was implemented using 3-point Radau collocation, with total simulation time (T) of 1s and 100 finite elements. The optimization was performed in GAMS using the CONOPT solver and the complementarity constraints were formulated using the penalty method. To assess accuracy, a hybrid-dynamic simulation was also implemented in Matlab using the ode45 solver (accuracy $1e-12$).

The results show that lowering the time-step bound increases the accuracy of the solution and in the limit the trajectory approaches the hybrid system trajectory, as seen in Fig. 3. Even in the worst case here, the impact event duration is around 2.5ms (and the other examples are all less than 1ms). The system is able to capture the sequential-but-simultaneous impact and liftoff transitions [25, Thm. 8], lasting one finite element each, however the liftoff is slightly delayed compared to the hybrid solution.

B. Double Pendulum with Hard-Stops Swing-Up

The second test system is a double pendulum with hard stops in the center joint, Fig. 4. For this system we compare the solving time and accuracy of three- and five-point Radau collocation algorithms (R3 and R5) to implicit Euler (IE) [1] and variational integration (VI) [16]. Using a torque τ_c acting at the base of the first link, the pendulum must swing itself up to a stationary vertical configuration.

The second link is constrained with hard contacts to swing only within $\pi/4$ radians relative to the first link. The following constraints assure that the rebound torque λ_r can only act when

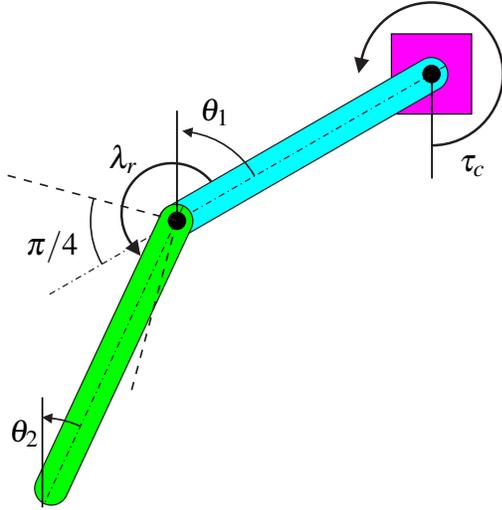


Fig. 4. Double pendulum model with hard stops at the center joint.

the link hits these bounds:

$$\alpha_{up}^+, \alpha_{up}^-, \alpha_{lo}^+, \alpha_{lo}^-, \lambda_r^+, \lambda_r^- \geq 0 \quad (23)$$

$$\alpha_{up}^+ - \alpha_{up}^- = \frac{\pi}{4} - (\theta_2 - \theta_1) \quad (24)$$

$$\alpha_{lo}^+ - \alpha_{lo}^- = -\frac{\pi}{4} - (\theta_2 - \theta_1) \quad (25)$$

$$\alpha_{up}^+ \lambda_r^- = 0 \quad (26)$$

$$\alpha_{lo}^- \lambda_r^+ = 0 \quad (27)$$

Solving this trajectory optimization problem is challenging for most NLP solvers. One strategy for improving the convergence rate is to provide a feasible (or close to feasible) initial solution [35]. As such, this problem was solved in two sequential stages: first solving for a feasible problem (no cost function) and then using this solution as a seed to the full problem to optimize the cost function,

$$g(z) = \sum_{i=1}^N \tau_{c,i}^2 h_i, \quad (28)$$

The active set solver CONOPT was utilized and the penalty method (17) was used to formulate the complementarity problem. The state variables were initialized with random values between $-\pi$ and π , while all other variables were set to a fixed nonzero value (0.01). Each algorithm was tested with 600 random seeds for three different problem sizes: $N=50$, 100, and 200 elements. Further tests using 600 and 1000 elements were run in the case of the IE algorithm, to correspond to the combined number of collocation points ($N \times K$) for $N=200$ with R3 and R5 (since the dynamics are evaluated at $N \times K$ points). The maneuver was executed in approximately 2 seconds, with the h_i allowed to vary within ± 20 percent of $2/N$ seconds.

The accuracy of the solutions was tested by comparing the state at each collocation point to the value generated by integrating the dynamic equations from the start of the finite element using MATLAB's `ode45` solver with $1e-12$ accuracy [29]. Since the VI method does not explicitly use instantaneous velocities, these values were approximated by calculating the

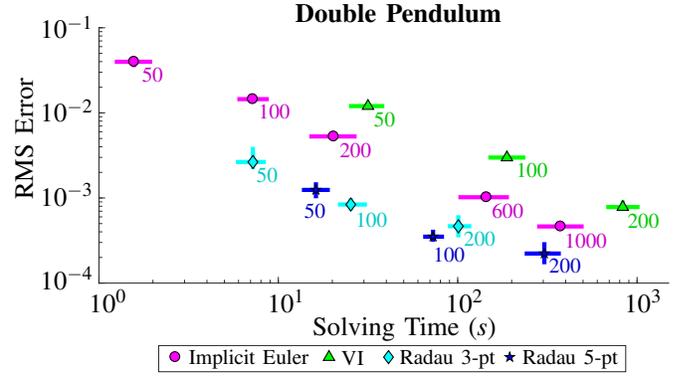


Fig. 5. Error and solving times for pendulum swing-up trajectories in all data sets. Bars indicate the inter-quartile range for 600 trials. Note that extra tests for $N=600$ and 1000 for implicit Euler are still not as accurate as $N=100$ Radau 5-point and are also much slower.

generalized momentum p from the discrete Lagrangian L_d as in [36]:

$$p_i = D^2 L_d(q_{i-1}, q_i) + \tau_{c,i} + \tau_{r,i}, \quad (29)$$

where, D^2 is the second derivative with respect to time. The velocities can then be evaluated using $p = \frac{\delta L}{\delta \dot{q}}$.

Each trial was run using four cores on a 32 core PC (Intel Xeon 2.2 GHz, 32 GB RAM). The median values and inter-quartile range for error and solving time for all problem are shown in Fig. 5. These results show that the IE and VI algorithms require more elements and longer solving times to achieve comparable accuracy to the Radau method (however see Sec. IV for a discussion of when a variational integrator may be warranted). For example, note that the $N=100$ R5 method is both faster and more accurate than either $N=600$ or $N=1000$ for IE (both of which evaluate the dynamics at more points than the $N \times K=500$ points with R5).

C. Biped

One of the primary motivations for this work is to generate long-time-horizon optimal motions with legged robots. Here, we optimize a biped model (based on [37]) to perform a 10m run. However, we do not prescribe periodicity nor contact order. In addition to the contact points at the toes, we also include contact points at the heels, for 4 total contact points. The model is constrained to start and end standing upright and at rest with the terminal condition being $x(t_f) = 10m$ with the objective function of energy minimization as in (28). A 1 m gap is also included, which the robot is required to negotiate.

The problem is discretized using $N = 300$ elements, $K = 3$ Radau collocation, and solved using IPOPT in GAMS. The complementarity constraints were formulated using the ε -relaxation method. Due to the four contact points and long-time horizon, this problem is significantly more challenging to solve the previous examples having around 90,000 variables and 100,000 constraints. We employ two stages of initialization: First, the problem of finding a feasible (fixed cost) and ε -relaxed ($\varepsilon = 10$) trajectory is solved with fixed time-steps. This trajectory is then used to seed the full problem, where the h_i are allowed to vary within ± 50 percent of T/N . Total

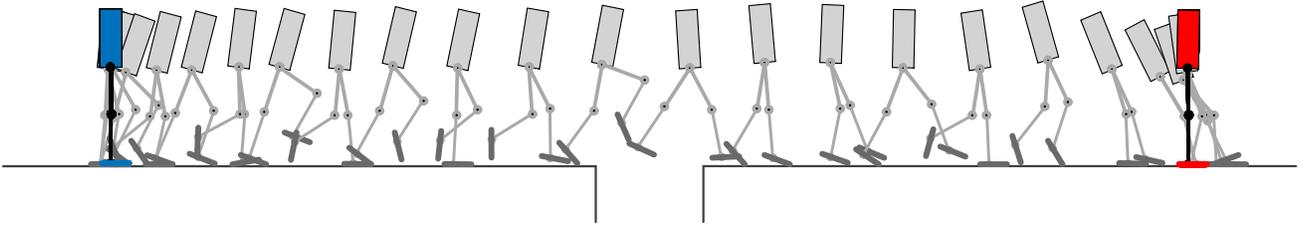


Fig. 6. Animation of the resulting trajectory showing that the model starts and ends at rest while traversing 10m and negotiating a gap in the floor.

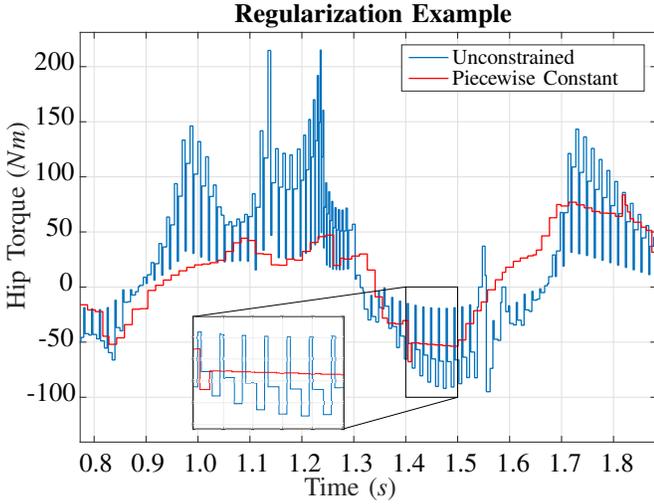


Fig. 7. A comparison between the unconstrained (changing at each collocation point) and piecewise constant (fixed over an entire finite element) torque is depicted. This illustrates the need for regularization of the control input to prevent oscillation and aid convergence.

convergence time was about 3 hours on the same PC described in III-B.

The resulting solution (Fig. 6) demonstrates that the method is able to generate motion plans without prescribing contact sequences for a multi-contact model. For comparison, an implicit Euler trajectory (see supplementary video) was also generated, however the resulting motion is quite unnatural, exhibiting excessive chattering and floating behaviour. As with the previous example, implicit Euler and 3-point Radau collocation were compared using `ode45` in Matlab and the higher order collocation improved the RMS error from 1.12×10^{-2} to 2.04×10^{-4} .

To illustrate the need for input torque regularization, we re-ran the same optimization and allowed the input torque to be unconstrained within the finite element (i.e. vary at each collocation point within the element). This nearly doubled the convergence time over the regularized (piecewise-constant over the finite element) torque version. Additionally, as is evident by Fig. 7, without regularization the torque oscillates, which would be undesirable when implementing this trajectory on a robot.

IV. DISCUSSION & FUTURE WORK

This paper presents a method for contact-implicit trajectory optimization which utilizes higher order orthogonal collocation to obtain a more accurate representation of the

dynamics than previous methods. To avoid the problems of discontinuities in the smooth representation [14, 19–21], we enforce mode changes to only occur at the mesh points. In contrast to the limitations of time-stepping simulation, where “if no collision detection is performed, the integration method cannot exceed order one,” [14], in the context of trajectory optimization where the time-steps are not fixed, collision detection is in fact being performed by the optimizer, allowing for higher order integration that approximates an event-driven simulation.

Transitioning from time-stepping to event-driven as the underlying simulation for the contact-implicit trajectory optimization is well justified in many robotic systems where the frequency of contact changes is relatively low (as opposed to, e.g., a billiard simulation). Because of this, the higher-order collocation allows for fewer finite elements to be utilized without loss of accuracy. Indeed, [30] notes that the time-stepping scheme, “is of order 1 therefore not very accurate unless h is decreased a lot ... it should therefore be preferred for systems with a lot of events only”. However, in the relaxed formulation presented here the impact events are stretched over the duration of a full finite element (as in time-stepping schemes). As such this method maintains the ability to model systems with many near-simultaneous impacts, with the same loss of precision that time-stepping incurs in isolating the effects of the separate events. The main disadvantage of our method (like other contact-implicit methods) is that it is not suited for cases with a large number of interacting bodies as each possible interaction would require a dedicated set of complementarity constraints, thereby increasing the problem size considerably [1].

One improvement of the presented method would be to use a more expressive control basis. In this formulation we have utilized piecewise-constant inputs (within the finite element) as a means to regularize and aid convergence of the singular control problems. Other input control profiles that vary over the finite element, but still provide the regularization to avoid the problems shown in Fig. 7, would improve the performance, e.g. as in [38].

Another interesting future direction of this work would be to combine the variational methods described by [16] with orthogonal collocation. This can be achieved by discretizing the Lagrangian using Radau collocation, thereby increasing the accuracy while maintaining the attractive energy preservation properties of the variational methods [39].

ACKNOWLEDGMENT

The authors would like to thank Joseph Norby, Zachary Manchester, and Scott Kuindersma for their helpful discussions.

REFERENCES

- [1] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.
- [2] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 43, 2012.
- [3] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Review*, vol. 59, no. 4, pp. 849–904, 2017.
- [4] A. Hereid, C. M. Hubicki, E. A. Cousineau, and A. D. Ames, "Dynamic humanoid locomotion: A scalable formulation for HZD gait optimization," *IEEE Transactions on Robotics*, 2018.
- [5] M. Posa, S. Kuindersma, and R. Tedrake, "Optimization and stabilization of trajectories for constrained dynamical systems," in *IEEE International Conference on Robotics and Automation*, May 2016, pp. 1366–1373.
- [6] K. Wampler and Z. Popović, "Optimal gait and form for animal locomotion," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 60:1–60:8, August 2009.
- [7] D. Pardo, M. Neunert, A. Winkler *et al.*, "Hybrid direct collocation and control in the constraint-consistent subspace for dynamic legged robot locomotion," in *Proceedings of Robotics: Science and Systems*, Cambridge, Massachusetts, July 2017.
- [8] 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*, vol. 3, no. 3, pp. 1560–1567, 2018.
- [9] B. Aceituno-Cabezas, C. Mastalli, H. Dai *et al.*, "Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2531–2538, 2018.
- [10] S. Kuindersma, R. Deits, M. Fallon *et al.*, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2016.
- [11] W. Xi, Y. Yesilevskiy, and C. D. Remy, "Selecting gaits for economical locomotion of legged robots," *The International Journal of Robotics Research*, vol. 35, no. 9, pp. 1140–1154, 2016.
- [12] Y. Yesilevskiy, W. Yang, and C. D. Remy, "Spine morphology and energetics: how principles from nature apply to robotics," *Bioinspiration & Biomimetics*, vol. 13, no. 3, p. 036002, 2018.
- [13] S. Shield and A. Patel, "Balancing stability and maneuverability during rapid gait termination in fast biped robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 4523–4530.
- [14] W. Xi and C. D. Remy, "Optimal gaits and motions for legged robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 3259–3265.
- [15] D. Stewart and J. C. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with coulomb friction," in *IEEE International Conference on Robotics and Automation*, vol. 1, 2000, pp. 162–169.
- [16] Z. Manchester and S. Kuindersma, "Variational contact-implicit trajectory optimization," in *International Symposium on Robotics Research*, 2017, pp. 1–16.
- [17] A. Werner, W. Turlej, and C. Ott, "Generation of locomotion trajectories for series elastic and viscoelastic bipedal robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2017, pp. 5853–5860.
- [18] K. Chao and P. Hur, "A step towards generating human-like walking gait via trajectory optimization through contact for a bipedal robot with one-sided springs on toes," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 4848–4853.
- [19] M. Anitescu, "A fixed time-step approach for multibody dynamics with contact and friction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2003, pp. 3725–3731.
- [20] M. Anitescu and G. D. Hart, "A constraint-stabilized time-stepping approach for rigid multibody dynamics with joints, contact and friction," *International Journal for Numerical Methods in Engineering*, vol. 60, no. 14, pp. 2335–2371, 2004.
- [21] V. Acary and B. Brogliato, *Numerical methods for nonsmooth dynamical systems*. Springer-Verlag, 2008.
- [22] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, pp. 338–342, 1987.
- [23] L. T. Biegler, *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. SIAM, 2010, vol. 10.
- [24] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010, vol. 19.
- [25] A. M. Johnson, S. E. Burden, and D. E. Koditschek, "A hybrid systems model for simple manipulation and self-manipulation systems," *International Journal of Robotics Research*, vol. 35, no. 11, pp. 1354–1392, September 2016.
- [26] D. Ralph and S. J. Wright, "Some properties of regularization and penalization schemes for MPECs," *Optimization Methods and Software*, vol. 19, no. 5, pp. 527–556, 2004.
- [27] T. Hoheisel, C. Kanzow, and A. Schwartz, "Theoretical and numerical comparison of relaxation methods for mathematical programs with complementarity constraints," *Mathematical Programming*, vol. 137, no. 1–2, pp. 257–288, 2013.
- [28] R. Fletcher and S. Leyffer, "Solving mathematical programs with complementarity constraints as nonlinear programs," *Optimization Methods and Software*, vol. 19, no. 1, pp. 15–40, 2004.
- [29] B. Baumrucker and L. Biegler, "MPEC strategies for optimization of a class of hybrid dynamic systems," *Journal of Process Control*, vol. 19, no. 8, pp. 1248–1256, 2009.
- [30] B. Brogliato, A. Ten Dam, L. Paoli *et al.*, "Numerical simulation of finite dimensional multibody nonsmooth mechanical systems," *Applied Mechanics Reviews*, vol. 55, no. 2, pp. 107–150, 2002.
- [31] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.
- [32] A. Drud, "A system for large scale nonlinear optimization, tutorial for CONOPT subroutine library," *AARKI Consulting and Development A/S*, 1995.
- [33] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM review*, vol. 47, no. 1, pp. 99–131, 2005.
- [34] M. R. Bussieck and A. Meeraus, "General algebraic modeling system (GAMS)," in *Modeling languages in mathematical optimization*. Springer, 2004, pp. 137–157.
- [35] S. M. Safdarnejad, J. D. Hedengren, N. R. Lewis, and E. L. Haseltine, "Initialization strategies for optimization of dynamic systems," *Computers & Chemical Engineering*, vol. 78, pp. 39–50, 2015.
- [36] A. Stern and M. Desbrun, "Discrete geometric mechanics for variational time integrators," in *ACM SIGGRAPH 2006 Courses*, 2006, pp. 75–80.
- [37] G. Schultz and K. Mombaur, "Modeling and optimal control of human-like running," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 5, pp. 783–792, 2010.
- [38] W. Chen and L. T. Biegler, "Nested direct transcription optimization for singular optimal control problems," *AICHE Journal*, vol. 62, no. 10, pp. 3611–3627, 2016.
- [39] J. E. Marsden and M. West, "Discrete mechanics and variational integrators," *Acta Numerica*, vol. 10, pp. 357–514, 2001.