

A Direct Method for Trajectory Optimization of Rigid Bodies Through Contact *

Michael Posa, Cecilia Cantu, and Russ Tedrake
Computer Science and Artificial Intelligence Lab
Massachusetts Institute of Technology
Cambridge, MA 02139
{mposa, ceci91, russt}@mit.edu

September 4, 2013

Abstract

Direct methods for trajectory optimization are widely used for planning locally optimal trajectories of robotic systems. Many critical tasks, such as locomotion and manipulation, often involve impacting the ground or objects in the environment. Most state-of-the-art techniques treat the discontinuous dynamics that result from impacts as discrete modes and restrict the search for a complete path to a specified sequence through these modes. Here we present a novel method for trajectory planning of rigid body systems that contact their environment through inelastic impacts and Coulomb friction. This method eliminates the requirement for *a priori* mode ordering. Motivated by the formulation of multi-contact dynamics as a Linear Complementarity Problem (LCP) for forward simulation, the proposed algorithm poses the optimization problem as a Mathematical Program with Complementarity Constraints (MPCC). We leverage Sequential Quadratic Programming (SQP) to naturally resolve contact constraint forces while simultaneously optimizing a trajectory that satisfies the complementarity constraints. The method scales well to high dimensional systems with large numbers of possible modes. We demonstrate the approach on four increasingly complex systems: rotating a pinned object with a finger, simple grasping and manipulation, planar walking with the Spring Flamingo robot, and high speed bipedal running on the FastRunner platform.

1 Introduction

Trajectory optimization is a powerful framework for planning locally optimal trajectories for linear or nonlinear dynamical systems. Given a control dynamical system, $\dot{x} = f(x, u)$, trajectory optimization aims to design a finite-time input trajectory, $u(t), \forall t \in [0, T]$, which minimizes some cost function over the resulting

*A preliminary version of this paper was presented at the 2012 Workshop on the Algorithmic Foundations of Robotics (WAFR) conference [27]

input and state trajectories. There are a number of popular methods for transcribing the trajectory optimization problem into a finitely parameterized nonlinear optimization problem (see [7]). Broadly speaking, these transcriptions fall into two categories: the *shooting* methods and the *direct* methods. In shooting methods, such as Differential Dynamic Programming (DDP) [20], the nonlinear optimization searches over (a finite parameterization of) $u(t)$, using a forward simulation from $x(0)$ to evaluate the cost of every candidate input trajectory. In direct methods, the nonlinear optimization simultaneously searches over parameterizations of $u(t)$ and $x(t)$; here no simulation is required and instead the dynamics are imposed as a set of optimization constraints, typically evaluated at a selection of *collocation* points [19]. Mixtures of shooting and direct methods are also possible, and fall under the umbrella of *multiple shooting*.

There are advantages and disadvantages to both direct and shooting methods, which are outlined here. For a more detailed comparison, see the survey article [6]. For many problems, direct methods enjoy a considerable numerical advantage over the shooting methods, which can be plagued by poorly conditioned gradients; for instance, a small change in the control input at $t = 0$ will often have a dramatically larger effect on the cost than a small change near time T . Direct methods can also be initialized with a guess for the state trajectory, $x(t)$, which may be easier to determine than an initial $u(t)$. A reasonable initial trajectory is generally helpful in avoiding problems with local minima. Since $x(t)$ is also parameterized, direct methods generate larger optimization problems but this increase in size is partially offset by the sparsity of the resulting problem, allowing efficient (locally optimal) solutions with large-scale sparse solvers such as SNOPT [18], and trivial parallel/distributed evaluation of the cost and constraints. Dimitrov et al. demonstrate the numerical benefits of sparse, direct optimization for MPC in [13]. We also note that throughout the optimization process, shooting methods that determine the state trajectory through simulation will always result in dynamically *feasible* trajectories. However, in the case of direct methods, solvers enforce the system dynamics through nonlinear constraints and must first converge to feasibility.

In this paper, we take advantage of the benefits of direct methods and consider the problem of trajectory optimization for rigid-body systems subject to elastic collisions and friction. This is an essential problem for robotics which arises in any tasks involving locomotion or manipulation. The collision events that correspond with making or breaking contact, however, greatly complicate the trajectory optimization problem as they result in large or impulsive forces and rapid changes in velocity. While it is possible to resolve contact through the use of continuous reaction forces like simulated springs and dampers, the resulting differential equations are typically stiff and require an extremely small time step, increasing the size and complexity of the problem [7]. For numerical efficiency, a preferred method is to approximate collisions as impulsive events that cause discontinuous jumps in velocity. A popular method for control of such systems is as a *autonomous hybrid* dynamical system that undergoes discontinuous switching (see [39]). The discrete transitions are fully autonomous as we can directly control neither the switching times nor the switching surface.

There are a number of impressive success stories for trajectory optimization in these hybrid models, for instance the optimization of a 3D running gait [30]. These results primarily use direct methods. But they are plagued with one major short-coming - the optimization is constrained to operate within an *a priori* specification of the ordering of hybrid modes. For a human running where motion

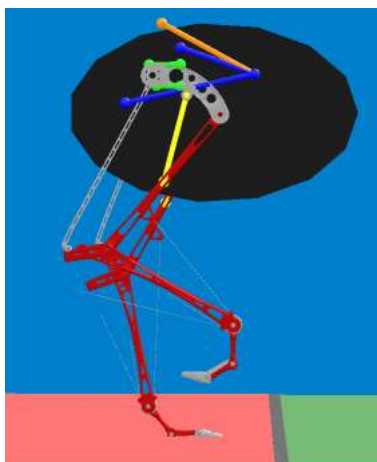


Figure 1: The bipedal FastRunner robot is designed to run at speeds of over 20 mph. Each leg has 5 degrees of freedom and multiple passive springs and tendons. The legs are driven at the hip to keep the leg mass as low as possible.

capture can provide a good initial guess on the trajectory, this may be acceptable. It is much more difficult to imagine a mode specification for a multi-fingered hand manipulating a complex object that is frequently making and breaking contact with different links on the hand. Perhaps as a result, there is an apparent lack of planning solutions for robotic manipulation which plan *through* contact - most planners plan up to a pre-grasp then activate a separate, heuristic based, grasping controller. Indeed, the multi-contact dynamics engines used to simulate grasping [21, 23] do not use hybrid models of the dynamics, because the permutations of different possible modes grows exponentially with the number of links and contact points, and because hybrid models can be plagued by infinitely-frequent collisions (e.g., when a bouncing ball comes to rest on a surface). Instead, simulation tools make use of time-stepping solutions that solve contact constraints using numerical solutions to linear or nonlinear complementarity problems (LCPs and NCPs) [34, 4].

We demonstrate that it is possible, indeed natural, to fold the complementarity constraints directly into nonlinear optimization for trajectory design, resulting in a Mathematical Program with Complementarity Constraints (MPCC), or, equivalently, a Mathematical Program with Equilibrium Constraints (MPEC) [22]. While these are generally difficult to solve, significant research has been done in this area, and we leverage Sequential Quadratic Programming (SQP) techniques - a particular class of algorithms for solving general nonlinear programs that have been shown to be effective [3, 16]. Broadly speaking, SQP solves a sequence of quadratic programs which each approximate the original nonlinear program. The key to this formulation is in resolving the contact forces, the mode-dependent component of the dynamics in the traditional formulation, as additional decision variables in the optimization. We demonstrate that this is an effective and numerically robust way to solve complex trajectories without the need for a mode schedule.

Specifically, this work was motivated by the challenge of optimizing trajectory-

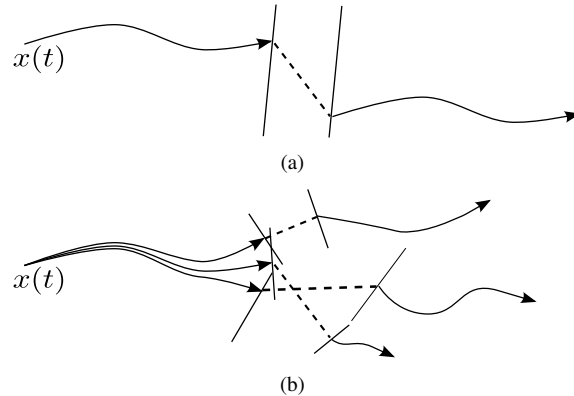


Figure 2: (a) Hybrid trajectories can be found by individually optimizing over the continuous dynamics of a specified mode sequence. Here, the system evolves continuously through one mode before striking a guard. The dashed line indicates the discontinuous jump from one mode to another before continuous evolution in the other mode begins. (b) When the hybrid transition map is more complex, the switching surfaces formed by multiple guards functions may lie close together and the set of possible mode sequences can be extremely large. In these cases, it is no longer trivial to specify the optimal mode sequence.

ries for a new running robot called “FastRunner” [12]. FastRunner, illustrated in Figure 1 is a bipedal robot concept designed to run at speeds over 20 mph and up to 50 mph. Most notably, FastRunner has a clever, but also complex, leg design with four-bar linkages, springs, clutches, hard joint stops, tendons and flexible toes. The planar FastRunner the model has 13 degrees of freedom, 6 contact points, 16 additional constraint forces, and only 2 actuators, and was beyond the scope of our previously existing trajectory optimization tools.

2 Background

2.1 The pitfalls of mode schedules

For simple hybrid systems, including point foot models of walking robots, trajectories of the hybrid system can be described by smooth dynamics up until a guard condition is met (e.g., the robot’s foot hits the ground), then a discontinuous jump in state-space, corresponding here to an instantaneous loss of velocity as energy is dissipated through an impulsive collision—followed by another smooth dynamical system, as cartooned in Figure 2(a). The discrete state, or mode, corresponds to the set of active contacts. The vector field, $\dot{x} = f(x, u)$, will also be mode dependent due to continuous forces exerted by sustained contact.

For a fixed mode schedule, direct methods for hybrid trajectory optimization proceed by optimizing each segment independently, with additional constraints ensuring that the segments connect to each other through the hybrid events.

However, when the dynamics are more complex, the geometric constraints imposed by the hybrid system become more daunting. The FastRunner model has a hybrid event every time that any of the ground contact points (on either foot) make or break contact with the ground. But the model also undergoes a hybrid transition every time that any one of the joints hits a joint-limit, and every time that any one of the ground reaction forces enters or leaves the friction cone (transitioning from a sticking to a sliding contact). Indeed, the number of possible hybrid modes of the system grows exponentially with the number of constraints. The geometry of the hybrid guards becomes very complex, as cartooned in Figure 2(b). In these models, small changes to the control input can result in a very different schedule of hybrid modes. Restricting the trajectory optimization search to the initial mode schedule can result in a very limited search and in failure to find a feasible trajectory that satisfies all of the constraints.

Despite the obvious limitation of requiring this mode schedule, it has proven surprisingly difficult to remove this assumption in the direct methods. Slight variations from the original sequence are possible if the formulation allows the time duration of individual modes to vanish, as in the work of Srinivasan and Ruina [32]. For problems with fewer possible modes, Wampler and Popović used outer optimization loops to determine the hybrid mode schedule [38]. In some cases, the combinatorial problem of solving for a mode schedule has been addressed by combinatorial planners; a variant of the Rapidly-Exploring Random Tree (RRT) algorithm was used in [31] to produce bounding trajectories for a quadruped over terrain. Methods for optimal control which approximate the global optimum, such as brute force methods based on dynamic programming, have also been applied [9], but are so far limited to low dimensional problems.

2.2 Contact Dynamics as a Complementarity Problem

In order to avoid the combinatorial explosion of hybrid models, simulation techniques in computer graphics and in grasping research use a different formulation of contact, summarized briefly here [37]. In this work, we will focus on systems comprised of multiple rigid bodies undergoing inelastic collisions and subject to Coulomb friction. Note that in some cases, like FastRunner, the rigid links composing the robot will be connected via (massless) passive elements like springs or tendons, though this does not fundamentally change the structure of the equations of motion. For simplicity, we first discuss the forward dynamics of a rigid-body (e.g., with a floating base) subject to *frictionless* contact constraints, which can be written as the problem of finding the acceleration and contact force vectors:

$$\text{find } \ddot{q}, \lambda \tag{1}$$

$$\text{subject to } H(q)\ddot{q} + C(q, \dot{q}) + G(q) = B(q)u + J(q)^T \lambda, \tag{2}$$

$$\phi(q) \geq 0 \tag{3}$$

$$\lambda \geq 0 \tag{4}$$

$$\phi(q)^T \lambda = 0. \tag{5}$$

where $q \in \mathbb{R}^n$ is the vector of generalized coordinates, $H(\cdot)$ is the inertial matrix, $C(\cdot, \cdot)$ represents the Coriolis terms, $G(\cdot)$ the gravitational forces, and $B(\cdot)$ is the input mapping. Vector inequalities are to be interpreted as element-wise inequality

constraints. For m potential contacts, $\phi(q) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ where $\phi(q) \geq 0$ is an inequality representing a non-penetration constraint. Strict equality $\phi_i(q) = 0$ holds if and only if the i th contact is active. $\lambda \in \mathbb{R}^m$ represents the constraint forces acting along the surface normal, and $J(\cdot)$ represents the Jacobian projecting constraint forces into the generalized coordinates. In simple examples, such as contact between a point and a fixed surface, we will take $\phi(q)$ to be the signed distance between the point and surface and then we have $J(q) = \frac{\partial \phi(q)}{\partial q}$. Taken together, (3)-(5) form a complementarity constraint and ensure that the contact forces can be non-zero if and only if the bodies are in contact. It should be noted that (3) may be difficult to pose properly in the case where the non-penetrating configuration space is non-convex, although work has been done in this area regarding simulation [26]. Standard variations on this formulation can be made to also address frictional contact [37, 8].

The solutions to these dynamics are potentially complex, involving large impact forces occurring over very short time periods (e.g., at the moment of a new collision). In the limiting case of purely rigid bodies, the constraint forces, $\lambda(t)$, are often modeled with Dirac δ functions, as impulsive events, or through measure differential inclusions [8, 33, 2]. However, many of these complexities can be avoided by discretizing the system in time. Stewart and Trinkle introduced a time-stepping method that only considers the integral of contact forces over a period and so does not differentiate between continuous and impulsive forces [34].

For the Stewart and Trinkle time-stepping method, the dynamics and constraint terms can be evaluated at the known q, \dot{q} , allowing an Euler-approximation of the state at the next time step to be written as the solution to a linear set of equations subject to linear complementarity constraints, resulting in an LCP. 3D contact models also fit into the LCP framework, although Coulomb friction cone must be approximated by a series of linear constraints. It has been proven that solutions exist to this LCP and, under reasonable conditions, that these solutions can be computed using pivoting methods like Lemke’s Algorithm or more general convex optimization algorithms [34, 4, 23]. Here, solving each LCP corresponds to simulating a single time step. This approach has proved to be remarkably efficient and can rapidly simulate systems with thousands of simultaneous contacts.

2.3 Related Work

A number of researchers are currently pursuing mode invariant trajectory optimization, as contrasted with the traditional hybrid systems based approach. In [5], Bernard et al. used the LCP formulation of contact to design trajectories of single body on a vibrating plate. Here, they optimize over a small set of parameters describing the oscillating behavior of the plate and the approach can best be described as a shooting method. Tassa and Todorov have also explored the use of stochastic complementarity for optimal control using DDP [36]. As this and similar approaches make local, gradient-based, improvements to a nominal trajectory, they will not naturally discover new contact sequences not already present and so rely on the natural dynamics of the system to make contact. Mordatch et al. demonstrated contact invariant optimization of complex, lifelike behaviors of humanoid figures using relaxations of the contact complementarity constraints in [25]. To pose a simpler problem, Mordatch et al. assume that the limbs of the figures are massless; an assumption we do not make in the current paper. The work of Erez and Todorov

optimized a human running gait by smoothing the contact dynamics to make use of a custom inverse dynamics formulation [14]. In [24], Mordatch et al. include the contact forces as optimization parameters, similar to the work in this paper and our preliminary version [27]. Mordatch includes the feasibility of the contact forces as a penalty term in the optimization cost function, rather than as a direct constraint. Broadly speaking, compared to this paper, the work in [14, 25, 24] make varying relaxations of physical or dynamic constraints to pose more tractable (and unconstrained) optimization problems. Relaxing the constraints allows contact forces to act at a distance; while this may produce dynamically infeasible trajectories, it has a smoothing effect and provides gradient information that would otherwise be unavailable in typical shooting methods. Additionally, all three works make heavy use of inverse dynamics. For highly underactuated systems, like the planar FastRunner robot, arbitrary state or end effector trajectories are not dynamically feasible and inverse dynamics is generally less useful. Formal comparison of these various methods is difficult, as the field has not yet agreed upon a set of canonical, hard problems; this is an important goal for collaborative future

3 Approach

Contact constraints formulated using the complementarity conditions fit naturally into the direct formulation of trajectory optimization. Rather than solving the LCP for the contact forces λ at each step, we directly optimize over the space of feasible states, control inputs, constraint forces, and trajectory durations. Treating the contact forces as optimization parameters is similar to how direct methods treat the state evolution implicitly. The number of parameters and constraints increases, but the problem is often better conditioned and more tractable to state of the art solvers. Where $g(\cdot, \cdot)$ and $g_f(\cdot)$ are the integrated and final cost functions respectively, the optimization problem can be written as

$$\underset{\{h, x_0, \dots, x_N, u_1, \dots, u_N, \lambda_1, \dots, \lambda_N\}}{\text{minimize}} \quad g_f(x_N) + h \sum_{k=1}^N g(x_{k-1}, u_k), \quad (6)$$

though any nonlinear objective based on $h, x_0, \dots, x_N, u_1, \dots, u_N, \lambda_1, \dots, \lambda_N$ is also allowed.

3.1 Optimization Constraints

This optimization problem is subject to constraints imposed by the manipulator dynamics and by rigid body contacts. To integrate the dynamics, both forwards and backwards Euler methods are equally applicable. Time-stepping simulation methods commonly use semi-implicit methods, but the dynamics constraints in our optimization problem are already fully implicit and so we chose backwards integration for added numerical stability. For ease of notation, we will write $H_k = H(q_k)$ and likewise for other matrix functions in the manipulator equations. Where h is the length of the time-steps and for $k = 1, \dots, N - 1$, the dynamics from (2) imply the constraints:

$$\begin{aligned} q_k - q_{k+1} + h\dot{q}_{k+1} &= 0 & (7) \\ H_{k+1}(\dot{q}_{k+1} - \dot{q}_k) + h \left(C_{k+1} + G_{k+1} - B_{k+1}u_{k+1} - J_{k+1}^T \lambda_{k+1} \right) &= 0. \end{aligned}$$

For notational simplicity, we first consider the case where the (frictional) contact dynamics are planar and later discuss the extension to 3D contacts. For a given contact point, write the contact force, $\lambda = [\lambda_x^+ - \lambda_x^- \quad \lambda_z]^T$ expressed in a reference frame with x tangent to the contact surface and z normal to the surface. Following the formulation of [34], we have split the tangential force into its positive and negative components and introduce the additional slack variable γ , which is generally equal to the magnitude of the relative tangential velocity at a contact. We then have the same set of unilateral and bilateral contact constraints:

$$\phi(q_k) \geq 0 \quad (8)$$

$$\lambda_{k,z}, \lambda_{k,x}^+, \lambda_{k,x}^-, \gamma_k \geq 0 \quad (9)$$

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

$$\gamma_k + \psi(q_k, \dot{q}_k) \geq 0 \quad (11)$$

$$\gamma_k - \psi(q_k, \dot{q}_k) \geq 0 \quad (12)$$

$$\phi(q_k)^T \lambda_{k,z} = 0 \quad (13)$$

$$\left(\mu \lambda_{k,z} - \lambda_{k,x}^+ - \lambda_{k,x}^- \right)^T \gamma_k = 0 \quad (14)$$

$$(\gamma_k + \psi(q_k, \dot{q}_k))^T \lambda_{k,x}^+ = 0 \quad (15)$$

$$(\gamma_k - \psi(q_k, \dot{q}_k))^T \lambda_{k,x}^- = 0. \quad (16)$$

where $\psi(q, \dot{q})$ is the relative tangential velocity at a contact. Taken together, (8)-(16) are a set of complementarity equations that describe inelastic impacts and a Coulomb coefficient of friction μ . In addition to preventing contact forces at a distance, these complementarity constraints enforce the friction cone and ensure that, if the contact is sliding, the tangential force properly lies on the edge of the cone and directly opposes the direction of motion. Together, these directly equate to the to the case structure of Coulomb friction where

$$\psi(q_k, \dot{q}_k) \neq 0 \Rightarrow \lambda_{k,x} = -\text{sgn}(\psi(q_k, \dot{q}_k)) \mu \lambda_{z,k}$$

$$\psi(q_k, \dot{q}_k) = 0 \Rightarrow \lambda_{k,x} \leq \mu \lambda_{z,k}.$$

In addition to expressing frictional contacts, we can also describe simple position constraints such as hard joint limits or kinematic loops in a similar manner. Here, λ is an internal torque or force acting directly on a joint. For example, if there is a physical stop enforcing the requirement that $q \leq q_{max}$, write

$$\phi(q_k) = q_{max} - q_k \geq 0 \quad (17)$$

$$-\lambda_k \geq 0 \quad (18)$$

$$\phi(q_k)^T \lambda_k = 0. \quad (19)$$

It is important to note the relative indexing of the complementarity and dynamical constraints. Over the interval $[t_k, t_{k+1}]$, the contact impulse can be non-zero if and only if $\phi(q_{k+1}) = 0$; that is, the bodies must be in contact at the end of the given interval. This allows the time-stepping integration scheme to approximate inelastic collisions where the interacting bodies stick together. This is not necessarily an appropriate approximation for bodies that may rapidly rebound off one

another, since any compliance must be modeled through a linkage in one of the bodies and the time step must be appropriately small.

3.2 Solving the Optimization Problem

The optimization problem (6), subject to the constraints in Section 3.1, forms an MPCC: a class of nonlinear programs that is generally difficult to solve due to the ill-posed nature of the constraints [22]. However, it is an area of optimization research that has garnered significant attention in recent years. There are a number of theoretical and practical results which we leverage here to ensure that our trajectory optimization problem is solvable with current techniques, particularly those used by the nonlinear solver SNOPT. Observing that, for vector-valued functions $G(x)$ and $H(x)$, many of our constraints are of the form

$$G(x) \geq 0 \tag{20}$$

$$H(x) \geq 0 \tag{21}$$

$$G(x)^T H(x) = 0. \tag{22}$$

To improve the convergence properties of the optimization routines, we can consider equivalent formulations of these complementarity conditions. Fukushima et al. propose an iterative method that sequentially tightens relaxations of the complementarity constraints [17]. In our work, we primarily adopt the scheme of Anitescu who proposed leveraging the elastic mode of SQP solvers like SNOPT to solve a set of similar, and equivalent constraints [3],

$$G(x) \geq 0 \tag{23}$$

$$H(x) \geq 0 \tag{24}$$

$$G_i(x)H_i(x) \leq 0, \tag{25}$$

where the last inequality is evaluated element-wise. Additionally, it was observed by Fletcher et al. in [16] that, since SQP iterations always satisfy linear constraints, the introduction of slack variables α and β can help avoid infeasible QP iterations:

$$\alpha, \beta \geq 0 \tag{26}$$

$$\alpha = G(x) \tag{27}$$

$$\beta = H(x) \tag{28}$$

$$\alpha_i \beta_i \leq 0. \tag{29}$$

In practice, these seemingly innocuous substitutions have greatly improved the speed and robustness of our optimization routines relative to our initial formulation, described in [27]. For more complex examples, we have also found it to be practically useful to temporarily relax the final constraint to $\alpha_i \beta_i \leq \varepsilon$ and solve a sequence of a few problems, starting with some $\varepsilon > 0$ and finishing with $\varepsilon = 0$ to achieve strict feasibility. This has the effect of allowing intermediate iterations to exert contact force at a small distance, and experimentally has improved the conditioning of the optimization problem and the quality of our solutions. This is similar in principle to existing approaches, like that in [17].

3.3 Extension to Three Dimensions

To handle three dimensional contacts, note that only the variables and constraints in Section 3.1 related to the frictional force λ_x and tangential velocity $\psi(q, \dot{q})$ are specific to the 2D case. One straightforward approach to extending to 3D would be to treat both λ_x and $\psi(q, \dot{q})$ as two-vectors, and write down a set of nonlinear constraints for Coulomb friction in the tangent plane, such as in [1]. However, to preserve the MPCC structure of our problem, we instead use a polyhedral approximation of the friction cone, as in [34]. Let D^i for $i = 1, \dots, d$ be unit vectors in \mathbb{R}^2 whose convex hull is the polyhedral approximation. Then, let $\lambda_x = \sum_i^d D^i \lambda_x^i$ be the net frictional force where each λ_x^i is a scalar. We replace the friction cone constraints (10)-(12) and (14)-(16) with

$$\lambda_{k,x}^i \geq 0 \quad (30)$$

$$\mu \lambda_{k,z} - \sum_i^d \lambda_{k,x}^i \geq 0 \quad (31)$$

$$\gamma_k + \psi(q_k, \dot{q}_k)^T D^i \geq 0 \quad (32)$$

$$\left(\mu \lambda_{k,z} - \sum_i^d \lambda_{k,x}^i \right)^T \gamma_k = 0 \quad (33)$$

$$\left(\gamma_k + \psi(q_k, \dot{q}_k)^T D^i \right)^T \lambda_{k,x}^i = 0, \quad (34)$$

where (30),(32), and (34) are repeated for all i . By increasing d , the size of the MPCC grows but the approximation can be made arbitrarily tight to the true friction cone.

3.4 Time Discretization

We also note here the role of the discrete time steps when resolving contacts. Since we use a time-stepping model, our approach makes no effort to determine the exact time that contact between bodies is made or broken. Impulsive and continuous forces are not treated independently and so we avoid the difficult and potentially combinatorial task of hybrid mode resolution. Instead, the constraint forces over the time step directly before a collision are precisely those required for the two bodies to be in contact. Additionally, since no force is permitted during the period when contact is being broken, there is the implicit requirement that take-off exactly coincide with one of the discrete time intervals. While it is common for numerical implementations of trajectory optimization to allow the overall duration of the trajectory to change, they typically do not adjust the individual time steps. Here this would result in an overly restrictive optimization problem that may exclude desirable trajectories. Overly simple parameterizations which use each time step duration as a parameter can have trivial or undesirable solutions (e.g. with many time steps having zero duration). One feasible approach is to create decision variables that divide each time step h into two periods. Where $\lfloor x \rfloor = \text{floor}(x)$, this can alternatively be expressed as having individual time steps h_k with pairwise constraints:

$$h_{2j-1} + h_{2j} = h_{2j+1} + h_{2j+2}, \quad j = 1, \dots, \left\lfloor \frac{N-3}{2} \right\rfloor, \quad (35)$$

In practice, these additional free parameters are useful in expanding the space of feasible solutions while still allowing for relatively large time steps. Since both state and constraint forces are solved implicitly, this program has a relatively large number of decision variables and constraints. However, as is typical in direct methods, this resulting program is generally sparse and so is suitable for implementation with sparse solvers.

For systems with a large number of bodies that could potentially interact with each other, there is the potential for k^2 collisions and $O(k^2)$ additional variables. For trajectory optimization, however, we are generally interested in problems where a robot is interacting with a limited set of environmental surfaces like the ground or an object for manipulation. In these cases, while we must treat a significant number of potential contacts, we avoid this additional quadratic complexity that might necessitate such a design by enumerating a small set of permitted contact body pairs.

4 Example Applications¹

Here, we have presented a method for generating locally optimal solutions to the task of mode agnostic trajectory optimization through contact. In this section, we apply this algorithm to four increasingly complex examples. These problems were solved on a standard desktop computer. While the solve time varied from problem to problem, the simpler examples completed in a few minutes or less in about one hundred major iterations of the solver, and the FastRunner trajectories took up to an hour to converge. As discussed above, the MPCCs are formulated to take advantage of the elastic mode in the SQP solver, although, in all cases, we achieve final convergence to a strictly feasible solution.

4.1 Finger Contact

Recent research by Tassa and Todorov used a DDP based approach to find a trajectory for the sample problem of a two link manipulator that must spin an ellipse. This is a simple example with three degrees of freedom and only one contact point, so there are only two possible modes [36]. However, it provided an early test for our methods. We constrained the system to start from rest, $\dot{q}_1 = 0, \dot{q}_2 = 0, \dot{q}_3 = 0$, and optimized for a quadratic cost on control input and velocity of the free ellipse:

$$g(x, u) = \sum_{k=1}^N \dot{q}_3^T Q \dot{q}_3 + u^T R u \quad (36)$$

The parameters for size and mass and for the cost function were chosen to directly parallel the previous work by Tassa and Todorov [36]. Our approach succeeded in quickly finding a locally optimal trajectory. As we increased the overall duration of the trajectory, the optimization process found an increasing number of flicking motions where, after making contact, it drew the finger back up to make another pass. Additionally, Tassa and Todorov note that the effect of gravity was required to pull the manipulator into contact with the ellipse in order for the optimization process to discover the possibility of contact. Our approach does not have this limitation. If we eliminate gravity from the system, even given an initial

¹A brief movie of the trajectories generated here can be found in Extension 1.

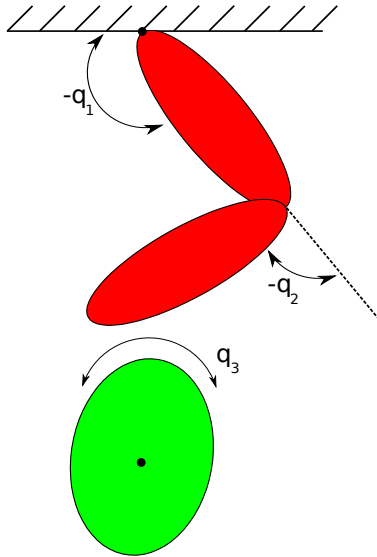


Figure 3: The two link finger, shown above, is fully actuated and makes contact with the unactuated third ellipse to drive it about its axis. Here, $\phi(q)$ is the shortest distance between the distal finger and the free ellipse.

trajectory that starts at rest with $u(t) = 0$, our methods successfully initiate contact between the manipulator and ellipse. Both of these results speak to the capability of our algorithm to actively identify a mode schedule that is not forced by the initial trajectory or the system’s dynamics.

4.2 Simple Manipulation

Given the complexity of manipulator problems, trajectory optimization usually involves dividing the planner into two parts: planning the motion to the object through unobstructed space and then subsequently planning the grasp. During grasp planning, specifying a mode schedule would require determining the order in which the manipulator fingers should interact with the object. However, it is clear that in many situations, the precise order is not important so long as a proper grasp is ultimately achieved. Thus, an optimization technique that does not require *a priori* specification of contact order is far more appropriate for these types of problems. Furthermore, some grasp planners neglect the dynamics of manipulated objects, essentially treating them as fixed to the manipulator. The method presented here fully accounts for the dynamical properties of the manipulated object throughout optimization of the entire trajectory. The following example consists of a planar manipulator tasked with grasping a circular object and lifting it into the air. We model the gripper with three contact points and five actuated joints, shown in Figure 4. Here, we desire to minimize the overall effort while moving the ball to a strictly specified goal location.

There are, of course, a number of different ways in which to precisely specify this optimization problem. Since we use a local method, the problem definition

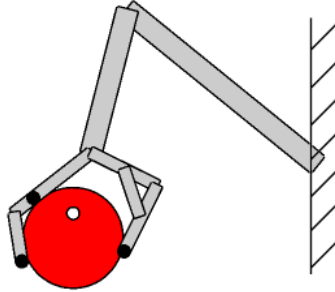


Figure 4: A simple planar gripper was modeled with five actuated joints and three contact points, shown as black dots. Both the ball and the three contact points could also interact with the ground, where the ball is initially resting, resulting in seven possible contacts.

and initial trajectories can have a significant impact on the final result. We choose an approach similar to standard grasp planning by specifying an intermediate state where the manipulator fully grasps the ball before attempting to raise it and we additionally specify the final state in terms of both ball and manipulator. For an initial trajectory, we construct a simple, three point linear interpolation between the initial, intermediate, and final states where the intermediate state is the grasp chosen by the user. Additionally, the optimization problem is initialized with all zeros for both control inputs and contact forces. Furthermore, we are able to require that the trajectory achieve a force closure on the ball, before attempting to raise it. With this simple problem formulation, our algorithm quickly converges to a trajectory that grasps and raises the ball. If, instead, we were to only specify a fixed initial state with the manipulator held above the ball, the algorithm could produce interesting motions that, while locally optimal, may not be what we had desired. For instance, using a single finger to hit the ball into the air satisfies this problem statement.

Since they are included as optimization parameters, the cost function and constraints can be easily modified to explicitly include the contact forces. For example, to handle the object gently, we could minimize the total contact forces between fingers and ball or even prohibit these forces from crossing a specified threshold.

4.3 Spring Flamingo Walking Gait

To analyze a more realistic system, we tested our methods on a planar simulation of the Spring Flamingo robot [29]. On Spring Flamingo, each leg has three actuated joints (hip, knee, and ankle) and there are contact points at the toe and heel of each foot. Many hybrid walking models use a constrained form of the dynamics, where a foot in contact with the ground is treated as a pin joint. Here, however, we deal with the full constrained dynamics where the body of the robot is modeled as a *floating base* parameterized by the variables (x, y, θ) which represent the planar position and pitch of the robot. Periodic constraints were used to generate a cyclic walking gait and the trajectory was optimized for mechanical cost of trans-

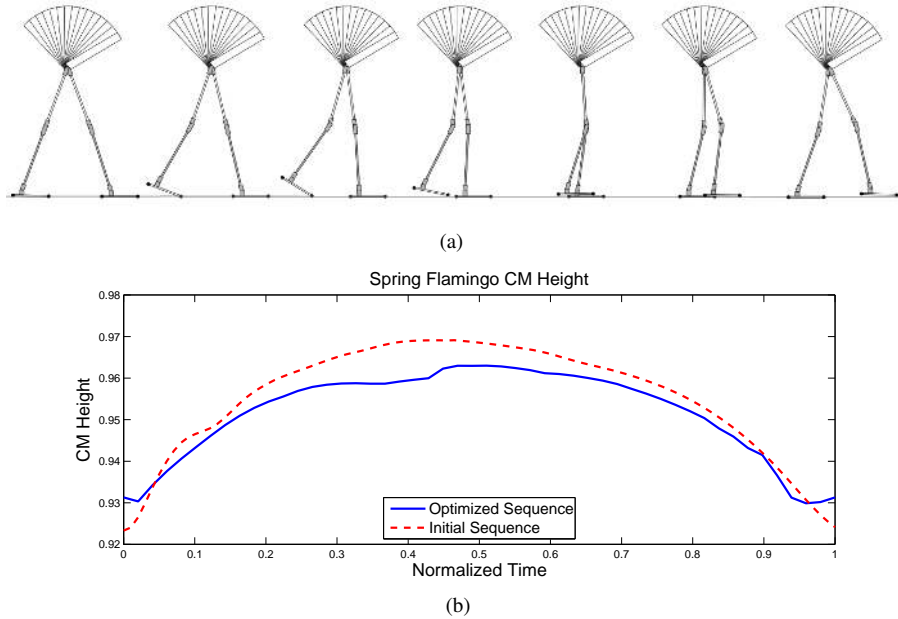


Figure 5: (a) A walking gait for Spring Flamingo that minimizes mechanical cost of transport. To generate this trajectory, the height of the swing foot was not considered, so the solution is a minimalist trajectory with very little ground clearance. (b) The height of the center of mass over the optimized and initial trajectories is plotted over the sequence. The optimal trajectory minimizes unnecessary vertical motion of the robot.

port. Cost of transport is a common, unitless indication of the energy consumption required for locomotion, and the “mechanical” cost of transport is computed using the total positive work done on the system independent of losses in the actuators or costs due to onboard electronics [11]. Where d is the total distance traveled, we write the cost as:

$$g(x, u) = \frac{1}{mgd} \sum_{k=1}^N \sum_i |\dot{q}_{k,i} u_{k,i}|. \quad (37)$$

Note that negative work, which could potentially be stored in an elastic element or harvested by regenerative braking, is simply treated here with an equivalent cost to positive work.

Since the solutions to the MPCC are local, our methods discovered a wide variety of feasible gaits that satisfied the general constraints dependent on the initial condition set. For instance, given the task of finding a periodic gait that travels a specific distance, hopping motions and gaits with relatively short or long strides are possible local solutions. In particular, the input $\lambda(t)$ and $x(t)$ sequences implicitly identify the nominal mode sequence of the initial guess. However, the solution is not restricted to the given ordering. Figure 6 shows the initial and optimized mode sequences of a particular Spring Flamingo gait. Here, the initial trajectory leads to a solution with a right-left walking gait but details such as independent heel strike

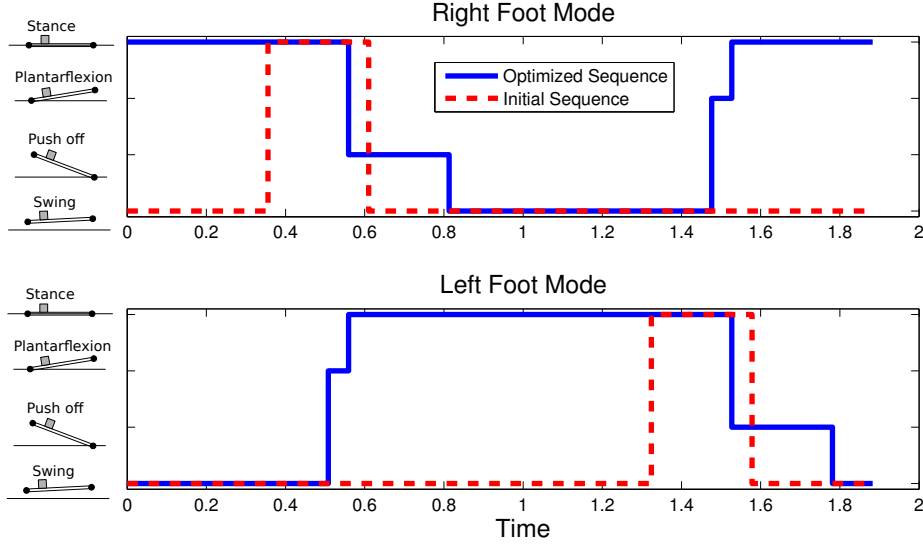


Figure 6: The optimized mode sequence of the left and right feet is plotted against time and the mode transitions are labeled. The SQP was initialized with a significantly different sequence, demonstrating the ability of the algorithm to independently plan through contact discontinuities. Note that, in this case, the locally optimal trajectory has distinct heel strike and heel off events.

and heel off were identified in the optimization process. To optimize for a cyclic gait, it is natural to write the periodicity constraint:

$$x_N = x_1. \quad (38)$$

Since we would prefer to search over the smaller space of a half gait, and we wish the robot to walk a minimum distance, we reformulate the periodic requirement to account for symmetry and add a unilateral constraint on stride length. Where q_l and q_r are the left and right joint vectors, respectively, d_{min} is the minimum stride length, and (x_{CM}, y_{CM}, θ) represents the position and orientation of the center of mass, we have:

$$\begin{bmatrix} y_{N,CM} \\ \theta_N \\ \dot{x}_{N,CM} \\ \dot{y}_{N,CM} \\ \dot{\theta}_N \\ q_{N,l} \\ q_{N,r} \\ \dot{q}_{N,l} \\ \dot{q}_{N,r} \end{bmatrix} = \begin{bmatrix} y_{1,CM} \\ \theta_1 \\ \dot{x}_{1,CM} \\ \dot{y}_{1,CM} \\ \dot{\theta}_1 \\ q_{1,r} \\ q_{1,l} \\ \dot{q}_{1,r} \\ \dot{q}_{1,l} \end{bmatrix}, \quad (39)$$

$$x_{N,CM} \geq x_{1,CM} + d_{min}. \quad (40)$$

With these linear constraints and given a nominal trajectory from Pratt's original work on the robot where the mechanical cost of transport was 0.18 [28], our

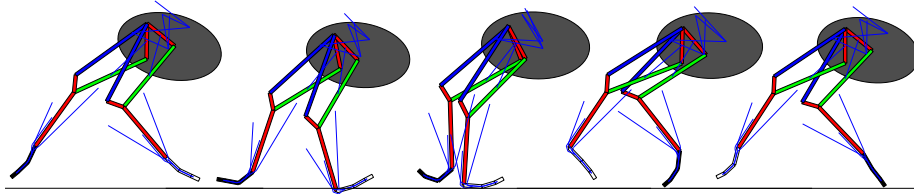


Figure 7: A generated trajectory for the FastRunner robot running at over 20 mph. The solid elements show the leg linkages and the thin lines indicate springs and tendons. Only the hip joints of the robot are actuated.

methods identified a periodic walking gait which reduced the cost to 0.04. It is important to note that this is merely the cost of the nominal gait as calculated from (37), and that stabilizing the gait in the presence of any disturbances or model error will result in a higher closed loop cost, even in simulation. Using native C++ code for rapid computation of rigid body dynamics, interfaced with a general purpose MATLAB framework, we are able to converge to solutions for the Spring Flamingo in under ninety seconds. The other examples discussed in this work are analyzed primarily in unoptimized MATLAB, and so are less relevant for comparison. This is a significant reduction in cost and corresponds to an impressive level of walking efficiency for a system with no passive elements to store and release energy. Figure 5(a) shows the optimized walking gait and the height of the center of mass (CM) throughout the trajectory, compared with that of the nominal gait. The optimal trajectory minimizes wasteful up and down motion of the CM. Note also that the foot swing height is very low to minimize any velocity at impact.

4.4 FastRunner Gait

The research behind this paper was motivated by the challenges posed by the FastRunner platform shown in Figure 1. For the previous examples, it is certainly possible to identify a desired mode sequence. This is a difficult task, however, for a system like FastRunner. A planar model of the robot has 13 degrees of freedom, including three articulated toe segments on each foot that can make or break contact with the ground. Additionally, there are a total of 16 unilateral joint limits, many of which are designed to be used while running at high speed. Scheduling the order of these contacts and joint limits is not practical.

Figure 7 shows a motion sequence of an optimized periodic running gait, averaging over 20 mph. This gait and the others mentioned here may also be seen in the video in Extension 1. As with Spring Flamingo, constraints (39) and (40) restricted the search space and this trajectory was optimized for mechanical cost of transport. Both the leg linkages and passive elements like springs and tendons are shown in the figure. For our model, we treat the system is a linkage of rigid bodies, where the passive elements are treated as massless. The complexity of the system and the stiffness of some of the springs posed additional problems for the optimization. In this case, additional linear constraints were useful in guiding the solver away from poorly conditioned or infeasible regions. This is typical for SQP methods, where the program can be difficult to solve if the local QP is a poor estimate of the true nonlinear program.

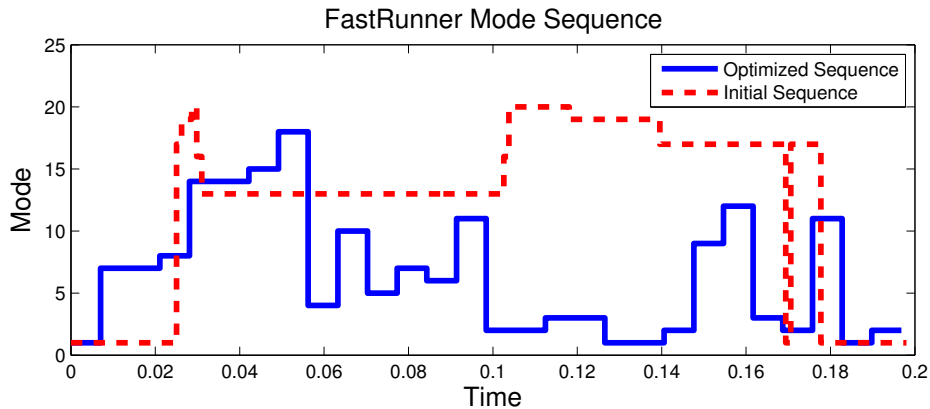


Figure 8: Out of more than 4 million possible discrete modes, the sequence for one locally optimal cyclic trajectory is shown. This sequence, which passes through 15 different modes, is compressed and plotted above against the largely distinct mode sequence of the initial trajectory.

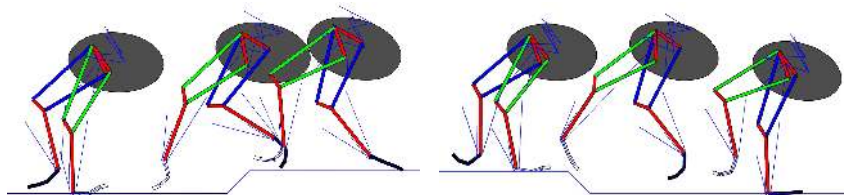


Figure 9: Constrained trajectory optimization can be used to generate gaits that deviate from the nominal motion. The images above show the robot ascending and descending 20 cm steps, more than double the height that has been achieved through passive stabilization alone.

With 22 discrete variables, there are over 4 million possible discrete modes for the FastRunner robot where each mode has a unique system of continuous dynamical equations. One possible mode sequence discovered by the optimization process is illustrated in Figure 8. The individual mode transitions shown occur when the contact state of one of the toes changes or when a joint limit becomes active or inactive. While the states of individual discrete variables, such as toe contacts, may overlap between the initial and optimal trajectories, the aggregate discrete states show almost no agreement. This speaks to the combinatorial complexity of planning a mode schedule for a system like FastRunner. Despite the complexity of the system, by taking advantage of the complementarity constraint formulation, our methods are now able to generate a locally optimal gait for FastRunner.

Our collaborators on the FastRunner project [12] have designed the robot to be open loop stable while tracking a simple sinusoidal gait. The design of the robot features an active clutch that connects a large suspensory spring to the knee joint during stance and then disconnects the spring during swing. It is critical that the clutch only be activated and deactivated when the spring is in the neutral

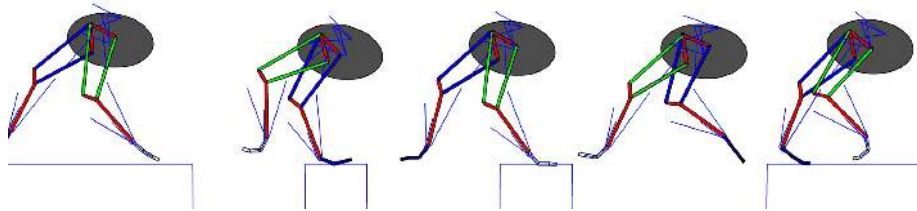


Figure 10: The nominal gait can also be modified with explicit foot placement constraints. Here, the robot must significantly alter its stride length to bridge the gaps shown above.

state and it has proven extremely difficult to design such a trajectory by hand. Such a constraint, however, fits naturally into our optimization algorithm. If c_k corresponds to the clutch activation at time t_k and $l(q_k)$ is the length of the spring, we then encode the two constraints:

$$c_k^2 - c_k = 0 \quad (41)$$

$$l(q_{k+1})(c_{k+1} - c_k) = 0 \quad (42)$$

The first equation ensures that c_k is binary, restricted to $\{0, 1\}$. The second describes the requirement that the clutch activation can only change if the spring is in the neutral position. With this new set of conditions, our optimization algorithm generates the complex trajectories required for FastRunner.

The trajectory optimization algorithm presented here allows us to synthesize efficient gaits for a wide range of different tasks. Of equal importance to generating a nominal gait is the ability to generate additional motions to handle atypical situations. For example, by modifying the model of the environment, our algorithm was successful at finding trajectories where the FastRunner robot must take 20 cm steps up or down, simulating running over rough (but known) terrain, all while running at high speeds. Figure 9 shows the robot mid-flight as it must step up and down. For these tasks, the initial and final states of the trajectory were constrained to precisely match the nominal running gait so that these motions can be smoothly strung together. We also applied our method to the task of explicitly modifying stride length, for situations where we must more tightly control foot placement. Figure 10 shows a stop-motion of the results of this optimization, where the small ledges force large deviations from the nominal stride length.

5 Discussion and Future work

In our approach, we write a complex MPCC that in practice has been tractable for state of the art solvers. Since the problem is non-convex, we are limited to locally optimal solutions. As is typical for non-convex problems, applying linear constraints on the decision variables to steer the solver away from singularities or other poorly conditioned regions can be critical to finding a desirable solution. In the examples above, this is typically done by eliminating obviously undesirable or infeasible regions of the joint space. As mentioned above, we have also found that certain, intermediate, relaxations of the complementarity conditions can greatly

improve the rate of convergence and reduce the likelihood of a poor, local solution. These relaxations are then tightened so the algorithm results in a strictly feasible trajectory. Other smoothing functions for nonlinear complementarity problems exist and have been used to directly solve these problems, such as the Fischer-Burmeister function [15] or the class of functions suggested in [10], and these functions may be applicable here as well.

As is mentioned above, throughout this paper we have solely dealt with inelastic collisions where the effective coefficient of restitution vanishes. While this is an appropriate assumption for the locomotion and manipulation examples explored here, there are other potential applications where the impacts are better modeled as partially elastic events. The work in [4] has developed LCP based simulation tools for multi-body contact and elastic collisions, and we believe our methods should extend to these areas as well.

Future work will also include extension of the methods described here to stabilize about the planned trajectory with a form of Model Predictive Control (MPC). Robust, contact invariant MPC of underactuated systems remains an open problem, although there has been significant recent work in this area, such as in [35]. We believe that, given a nominal trajectory, this approach can be adapted to solve the problem of real-time local control and that the controller will be capable of planning a finite-horizon path with a different mode sequence than that of the nominal trajectory.

6 Conclusion

To control highly nonlinear robotic systems through real-world environments, it is critical that we be able to generate feasible, high quality trajectories. Current, state of the art techniques struggle when presented with complex systems where the hybrid sequence is difficult to intuit. Here, we have presented a method for trajectory optimization through the discontinuities of contact that does not rely on *a priori* specification of a mode schedule. Our approach combines traditional, direct local control approaches with an complementarity based contact model into a single nonlinear program. By writing the dynamics and constraints without explicit reference to hybrid modes, we are now able to easily plan through the discontinuous dynamics of contact. Additionally, unlike with other methods, we do not require arbitrary or hand-tuned parameters nor do we rely on the passive system dynamics to generate a mode schedule. Once convergence is reached, the solution strictly satisfies all contact and dynamics constraints. We applied our method to four different systems, including the high dimensional FastRunner robot where we generated a high speed running gait. Future efforts will be focused on improving the convergence properties of this algorithm and extending it to real-time trajectory stabilization.

7 Funding

This work was supported by the Defense Advanced Research Projects Agency (DARPA) Maximum Mobility and Manipulation program [W91CRB-11-1-0001] and the National Science Foundation (NSF) [IIS-0746194, IIS-1161909, IIS-0915148].

References

- [1] V. Acary and B. Brogliato. *Numerical methods for nonsmooth dynamical systems: applications in mechanics and electronics*, volume 35. Springer, 2008.
- [2] V. Acary, B. Brogliato, A. Daniilidis, and C. Lemaréchal. On the equivalence between complementarity systems, projected systems and unilateral differential inclusions. 2004.
- [3] M. Anitescu. On using the elastic mode in nonlinear programming approaches to mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, 15(4):1203–1236, 2005.
- [4] M. Anitescu and F.A. Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics*, 14(3):231–247, 1997.
- [5] S. Berard, B. Nguyen, K. Anderson, and J.C. Trinkle. Sources of error in a simulation of rigid parts on a vibrating rigid plate. *ASME Journal of Computational and Nonlinear Dynamics*, 5(4):041003–1, 2010.
- [6] John T. Betts. Survey of numerical methods for trajectory optimization. *Journal of guidance, control, and dynamics*, 21(2):193–207, 1998.
- [7] John T. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming*. SIAM Advances in Design and Control. Society for Industrial and Applied Mathematics, 2001.
- [8] B. Brogliato. *Nonsmooth mechanics: models, dynamics, and control*. Springer Verlag, 1999.
- [9] Katie Byl and Russ Tedrake. Approximate optimal control of the compass gait on rough terrain. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2008.
- [10] C. Chen and O.L. Mangasarian. A class of smoothing functions for nonlinear and mixed complementarity problems. *Computational Optimization and Applications*, 5(2):97–138, 1996.
- [11] Steven H. Collins, Andy Ruina, Russ Tedrake, and Martijn Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307:1082–1085, February 18 2005.
- [12] S. Cotton, I. Oлару, M. Bellman, T. van der Ven, J. Godowski, and J. Pratt. Fastrunner: A fast, efficient and robust bipedal robot. concept and planar simulation. In *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012.
- [13] Dimitar Dimitrov, Alexander Sherikov, and Pierre-Brice Wieber. A sparse model predictive control formulation for walking motion generation. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.
- [14] T. Erez and E. Todorov. Trajectory optimization for domains with contacts using inverse dynamics. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4914–4919. IEEE, 2012.
- [15] A. Fischer. A special newton-type optimization method. *Optimization*, 24(3-4):269–284, 1992.

- [16] R. Fletcher, S. Leyffer, D. Ralph, and S. Scholtes. Local convergence of sqp methods for mathematical programs with equilibrium constraints. *SIAM Journal on Optimization*, 17(1):259–286, 2006.
- [17] M. Fukushima, Z.Q. Luo, and J.S. Pang. A globally convergent sequential quadratic programming algorithm for mathematical programs with linear complementarity constraints. *Computational Optimization and Applications*, 10(1):5–34, 1998.
- [18] Philip E. Gill, Walter Murray, and Michael A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005.
- [19] C. R. Hargraves and S. W. Paris. Direct trajectory optimization using nonlinear programming and collocation. *J Guidance*, 10(4):338–342, July-August 1987.
- [20] David H. Jacobson and David Q. Mayne. *Differential Dynamic Programming*. American Elsevier Publishing Company, Inc., 1970.
- [21] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2149–2154. IEEE, 2004.
- [22] Z.Q. Luo, J.S. Pang, and D.Ralph. *Mathematical programs with equilibrium constraints*. Cambridge University Press, 1996.
- [23] A.T. Miller and H.I. Christensen. Implementation of multi-rigid-body dynamics within a robotic grasping simulator. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 2262–2268. IEEE, 2003.
- [24] I. Mordatch, Z. Popović, and E. Todorov. Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 137–144. Eurographics Association, 2012.
- [25] I. Mordatch, E. Todorov, and Z. Popović. Discovery of complex behaviors through contact-invariant optimization. *To appear in ACM SIGGRAPH*, 31(4):43, 2012.
- [26] B. Nguyen and J. Trinkle. Modeling non-convex configuration space using linear complementarity problems. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2316–2321. IEEE, 2010.
- [27] Michael Posa and Russ Tedrake. Direct trajectory optimization of rigid body dynamical systems through contact. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*, Cambridge, MA, 2012.
- [28] Jerry Pratt. *Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots*. PhD thesis, Computer Science Department, Massachusetts Institute of Technology, 2000.
- [29] Jerry Pratt and Gill Pratt. Intuitive control of a planar bipedal walking robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1998.

- [30] G. Schultz and K. Mombaur. Modeling and optimal control of human-like running. *IEEE/ASME Transactions on Mechatronics*, 15(5):783–792, Oct. 2010.
- [31] Alexander Shkolnik, Michael Levashov, Ian R. Manchester, and Russ Tedrake. Bounding on rough terrain with the littledog robot. *The International Journal of Robotics Research (IJRR)*, 30(2):192–215, Feb 2011.
- [32] Manoj Srinivasan and Andy Ruina. Computer optimization of a minimal biped model discovers walking and running. *Nature*, 439:72–75, January 5 2006.
- [33] D.E. Stewart. Rigid-body dynamics with friction and impact. *SIAM Review*, 42(1):3–39, 2000.
- [34] D.E. Stewart and J.C. Trinkle. 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, 1996.
- [35] Y. Tassa, T. Erez, and E. Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4906–4913. IEEE, 2012.
- [36] Y. Tassa and E. Todorov. Stochastic complementarity for local control of discontinuous dynamics. In *Proceedings of Robotics: Science and Systems (RSS)*. Citeseer, 2010.
- [37] J.C. Trinkle, J.S. Pang, S. Sudarsky, and G. Lo. On dynamic multi-rigid-body contact problems with coulomb friction. *ZAMM-Journal of Applied Mathematics and Mechanics*, 77(4):267–279, 1997.
- [38] K. Wampler and Z. Popović. Optimal gait and form for animal locomotion. In *ACM Transactions on Graphics (TOG)*, volume 28, page 60. ACM, 2009.
- [39] Eric R. Westervelt, Jessy W. Grizzle, Christine Chevallereau, Jun Ho Choi, and Benjamin Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press, Boca Raton, FL, 2007.

A Index to Multimedia Extensions

The multimedia extensions to this article are at: <http://www.ijrr.org>.

Extension	Type	Description
1	Video	Animations of the various trajectories generated in Section 4.