

MIT Open Access Articles

*Direct Trajectory Optimization of Rigid
Body Dynamical Systems through Contact*

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Posa, Michael and Russ Tedrake. "Direct Trajectory Optimization of Rigid Body Dynamical Systems through Contact." Algorithmic Foundations of Robotics X: Proceedings of the Tenth Workshop on the Algorithmic Foundations of Robotics, edited by Emilio Frazzoli, et al., Springer, 2013: 527-542.

As Published: http://dx.doi.org/0.1007/978-3-642-36279-8_32

Publisher: Springer Berlin Heidelberg

Persistent URL: <https://hdl.handle.net/1721.1/124395>

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of use: Creative Commons Attribution-Noncommercial-Share Alike



Direct Trajectory Optimization of Rigid Body Dynamical Systems Through Contact

Michael Posa and Russ Tedrake

Abstract Direct methods for trajectory optimization are widely used for planning locally optimal trajectories of robotic systems. Most state-of-the-art techniques treat the discontinuous dynamics of contact 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 through contact that eliminates the requirement for an *a priori* mode ordering. Motivated by the formulation of multi-contact dynamics as a Linear Complementarity Problem (LCP) for forward simulation, the proposed algorithm leverages Sequential Quadratic Programming (SQP) to naturally resolve contact constraint forces while simultaneously optimizing a trajectory and satisfying nonlinear complementarity constraints. The method scales well to high dimensional systems with large numbers of possible modes. We demonstrate the approach using three increasingly complex systems: rotating a pinned object with a finger, 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, $\dot{x} = f(x, u)$, and an initial condition of the system $x(0)$, 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 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 [2]). Broadly speaking, these transcriptions fall into two categories: the *shooting* methods and the *direct* methods.

Michael Posa and Russ Tedrake
Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02193, e-mail:
{mposa@mit.edu, russt@mit.edu}

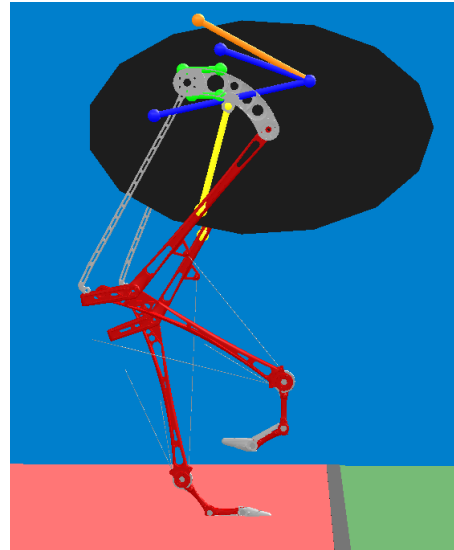
In shooting methods, the nonlinear optimization searches over (a finite parameterization of) $u(t)$, using a forward simulation from $x(0)$ to evaluate the cost of every candidate input trajectory. In direct methods, the nonlinear optimization simultaneously searches over parameterizations of $u(t)$ 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[9]. Mixtures of shooting and direct methods are also possible, and fall under the umbrella of *multiple shooting*.

Direct methods typically enjoy a considerable numerical advantage over the shooting methods. Shooting methods are plagued by poorly conditioned gradients; for instance, a small change in the input tape 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 an initial guess for the state trajectory, $x(t)$, which can often be easier to determine than an initial tape for $u(t)$. A reasonable initial tape is generally helpful in avoiding problems with local minima. The resulting optimization problems are also sparse, allowing efficient (locally optimal) solutions with large-scale sparse solvers such as SNOPT[8], and trivial parallel/distributed evaluation of the cost and constraints.

In this paper, we consider the problem of designing direct trajectory optimization methods for rigid-body systems with contact. 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, making trajectory optimization very inefficient. For numerical efficiency, a preferred method is to approximate contact as an *autonomous hybrid* dynamical system that undergoes discontinuous switching (see [20]). 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 an impressive 3D running gait[14]. 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 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 grasping controller. Indeed, the multi-contact dynamics engines used to simulate grasping[10, 11] 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

Fig. 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.



ball comes to rest on a surface). Instead, simulation tools make use of time-stepping solutions that solve contact constraints using numerical solutions to a linear complementarity problem (LCP) which can be solved with well known methods like Lemke’s Algorithm[17]. Significant additional work has been done to extend the original LCP methods to guarantee solutions to multi-body contact[1, 11].

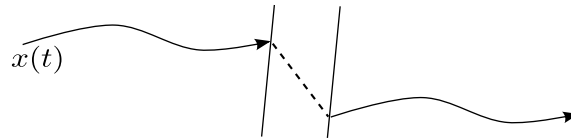
In many cases, solutions to the LCP for forward simulation can be found via the minimization of a convex quadratic program (QP). In this paper we demonstrate that it is possible, indeed natural, to fold the complementarity constraints directly into nonlinear optimization for trajectory design, which we solve here using sequential quadratic programming (SQP). The key trick is 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 trajectories for a new running robot called “FastRunner”[5]. 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 an exceptionally clever, but also exceptionally 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, and 16 additional constraint forces, and was beyond the scope of our previously existing trajectory optimization tools.

1.1 The pitfalls of mode schedules

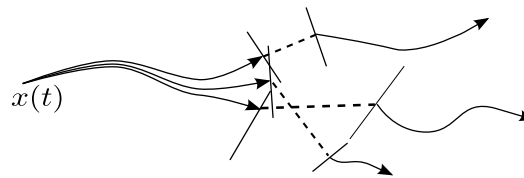
For simple hybrid systems, including point foot models of walking robots, trajectories of the hybrid system can be described by a smooth dynamics up until a guard condition is met (e.g., the robot’s foot hits the ground), then a discontinuous jump in state-space, corresponding here to an instantaneous loss of velocity as energy is dissipated into the ground—followed by another smooth dynamical system, as cartooned in Figure 2. For a fixed mode schedule, direct methods for hybrid trajectory optimization proceed by optimizing each segment independently, with additional constraints enforcing that the segments connect to each other through the hybrid events.

Fig. 2 Hybrid trajectories can be found by individually optimizing over the continuous dynamics of a specified mode sequence. The dashed line indicates the discontinuous jump from one mode to another.



However, when the dynamics get 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 rigid 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 3. In these models, small changes to the input tape 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 often in failure to find a feasible trajectory that satisfies all of the constraints.

Fig. 3 When the hybrid transition map of Figure 2 becomes increasingly complex, it is no longer trivial to specify the optimal mode sequence.



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 [16]. For problems with fewer possible modes, outer optimization loops have been used to determine the hybrid mode schedule, as in [19]. In some cases, the combinatorial problem of solving for a mode schedule have been addressed by combinatorial planners; a variant of the Rapidly-Exploring Random Tree (RRT) algorithm was used in [15] to produce bounding trajectories for a quadruped over terrain. Methods for optimal control which approximate the global optimal, such as brute force methods based on dynamic programming, have also been applied[3], but are so far limited to low dimensional problems.

1.2 Contact Dynamics as a Linear 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. The forward dynamics of a rigid-body (e.g., with a floating base) subject to contact constraints can be written as

$$\text{find } \ddot{q}, \lambda \quad (1)$$

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

$$\phi(q) \geq 0 \quad (3)$$

$$\lambda \geq 0 \quad (4)$$

$$\forall i, \phi_i(q)\lambda_i = 0 \quad (5)$$

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. $\phi(q) \geq 0$ represents a non-penetration constraint where the equality holds if and only if the two bodies are in contact, λ represents the constraint forces, and $J(\cdot) = \frac{\partial \phi}{\partial q}$ represents the Jacobian projecting constraint forces into the generalized coordinates. Vector inequalities are to be interpreted as element-wise inequality constraints. Equation 5, often referred to as a complementarity constraint, ensures that the contact forces can be non-zero if and only if the bodies are in contact. Solving these equations for \ddot{q} requires solving a Nonlinear Complementarity Problem (NCP).

The solution to these dynamics are potentially complex, involving high impact forces occurring across very short time periods (e.g., at the moment of a new collision). In the limiting case of purely rigid bodies, the constraint forces, $\lambda(t)$, must be modeled with the Dirac δ functions, or as hybrid impulsive events. However, many of these complexities can be avoided by discretizing the system in time. [17] 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.

For the Stewart and Trinkle time-stepping method, the manipulator dynamics and constraint terms can be evaluated at the known q, \dot{q} , allowing 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. For 3D models, the Coulomb friction cone must be approximated by a series of linear constraints, but otherwise still fits into the LCP framework. It has been proven that solutions exist to this LCP and, under reasonable conditions, can be computed using Lemke’s Algorithm or similar methods. Here, solving each LCP corresponds to simulating a single time step.

Despite the success in forward simulation, to our knowledge the LCP formulation has not been used in direct trajectory optimization. In related work, [18] explored the use of Stochastic Complementarity Problems in a shooting method. Their approach does not require an explicit mode schedule, but instead relies on the passive dynamics of the system to initiate contact. In cases where the simulated dynamics of the initial tape do not find the desired mode sequence, no gradient information will be available to guide the local search.

2 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.

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

Here, $g(\cdot, \cdot)$ and $g_f(\cdot)$ are the integrated and final cost functions respectively. 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. Unlike with forward simulation, our methods are already implicit and so backwards integration adds no computational cost. Given that we wish to use large time intervals, 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 dynamics. Where h is the length of the time-steps and for $k = 1, \dots, N - 1$, the manipulator dynamics from (2) imply the constraints:

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

For the rest of this paper, we will, for simplicity, only consider the case where the contact dynamics are planar. However, since we use the unsimplified nonlinear constraints, these methods easily extend to 3D contacts. For a given contact point,

write the contact force in a reference frame aligned to the contact surface, $\lambda = [\lambda_x \lambda_z]^T$. We then have the same set of unilateral and bilateral contact constraints as before:

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

$$\lambda_{k,z} \geq 0 \quad (9)$$

$$(\mu \lambda_{k,z})^2 - \lambda_{k,x}^2 \geq 0 \quad (10)$$

$$\phi(q_k) \lambda_{k,z} = 0 \quad (11)$$

$$\Psi(q_k, \dot{q}_k) \lambda_{k,z} = 0 \quad (12)$$

where $\Psi(q, \dot{q})$ is the relative velocity between two bodies that can make contact and Equation 12 represents an additional no-slip constraint.

Unlike with the task of pure simulation where these constraints and the dynamical constraints in (7) can be linearized about the initial state, here we must consider the higher order behavior and so we use the true formulation as a Nonlinear Complementarity Problem (NCP). Coulomb friction is described in (10), although any differentiable friction formulation could be used. We can construct an alternative formulation that allows for trajectories that violate the no-slip constraint by modeling sliding and kinetic friction. To do this, we introduce the slack variable $\gamma_k \geq 0$ that, when the lateral force is non-zero, represents the magnitude of the contact velocity. Additionally, substitute $\lambda_{k,x} = \lambda_{k,x}^+ - \lambda_{k,x}^-$ where $\lambda_{k,x}^+, \lambda_{k,x}^- \geq 0$. With these new contact parameters, replace (10) and (12) with:

$$\gamma_k + \Psi(q_k, \dot{q}_k) \geq 0 \quad (13)$$

$$\gamma_k - \Psi(q_k, \dot{q}_k) \geq 0 \quad (14)$$

$$\mu \lambda_{k,z} - \lambda_{k,x}^+ - \lambda_{k,x}^- \geq 0 \quad (15)$$

$$(\gamma_k + \Psi(q_k, \dot{q}_k)) \lambda_{k,x}^+ = 0 \quad (16)$$

$$(\gamma_k - \Psi(q_k, \dot{q}_k)) \lambda_{k,x}^- = 0 \quad (17)$$

$$(\mu \lambda_{k,z} - \lambda_{k,x}^+ - \lambda_{k,x}^-) \gamma_k = 0 \quad (18)$$

Simple position constraints, like joint limits, can be written in a similar manner. Here, λ is a torque or force acting directly on a joint. To express the requirement that $q \leq q_{max}$, write:

$$\phi(q_k) = q_{max} - q_k \geq 0 \quad (19)$$

$$\lambda_k \leq 0 \quad (20)$$

$$\phi(q_k) \lambda_k = 0 \quad (21)$$

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.

We also note here the role of the discrete time steps when resolving contacts. Our approach makes no effort to determine the exact time that contact between bodies is made or broken. The constraint forces over the time step directly before collision are precisely those required for the two bodies to be tangentially 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. Traditional optimal control approaches allow the overall duration of the trajectory to change, but not the individual time steps. This results in an overly restrictive optimization problem that may exclude desirable trajectories. One feasible solution is to create decision variables that divide each time step h into two periods. This can alternatively be expressed as having individual time steps h_k with pairwise constraints:

$$h_{2j-1} + h_{2j} = h_{2k+1} + h_{2j+2}, j = 1, \dots, \lfloor \frac{N-3}{2} \rfloor \quad (22)$$

In practice, these additional free parameters are useful in expanding the space of feasible solutions while still allowing for relatively large time steps. Unlike the tools designed for multi-body simulation, we still do not need to explicitly solve for the zero-crossings when two bodies make and break contact[1]. Additionally, 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, we avoid the combinatorial complexity of the number of potential contacts that might necessitate such a design.

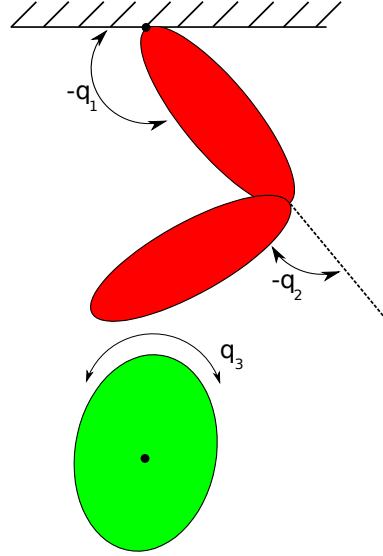
Since both state and constraint forces are solved implicitly, this program has a relatively large number of decision variables and constraints. However, as is typical in direct methods, this resulting NP is generally sparse and so is suitable for implementation with SNOPT[7].

3 Example Applications

3.1 *Finger Contact*

Recent research by Tassa and Todorov has explored the possibility of using stochastic complementarity with optimal control[18]. In this work, they use a Dynamic Programming 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 where the optimal mode schedule is immediately obvious. However, it provided an early test for our methods. We con-

Fig. 4 The finger, shown in red, is fully actuated and makes contact with the third ellipse to drive it about its axis. Here, $\phi(q)$ is the shortest distance between the distal finger and the free ellipse.



strained the system to start from rest, $q_1 = 0, q_2 = 0, q_3 = 0$, and optimized for a quadratic cost on control input and velocity of the free ellipse:

$$g(x, u) = \sum_{k=1}^N \dot{q}_3^T Q \dot{q}_3 + u^T R u \quad (23)$$

The parameters for size and mass and for the cost function were chosen to directly parallel the previous work by Tassa and Todorov. As we hoped, our approach succeeded in quickly finding a locally optimal trajectory. For $N = 20$, convergence is reached in roughly 30 seconds. 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 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 tape or the system's dynamics.

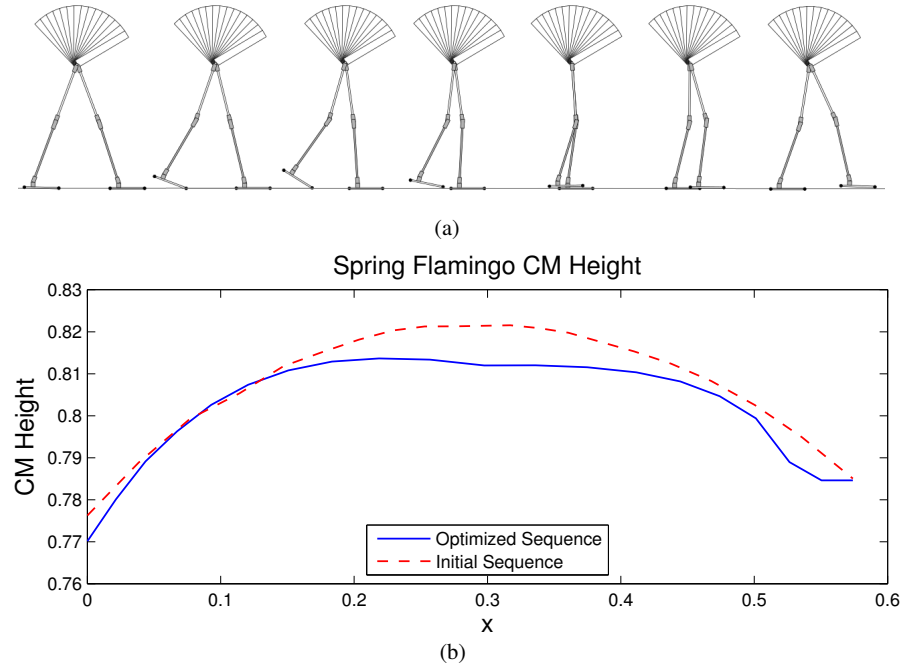


Fig. 5 (a) A walking gait for Spring Flamingo that minimizes mechanical cost of transport. To generate this trajectory, the height of the swing foot was not considered, so the solution is a minimalist trajectory with very little ground clearance. (b) The height of the center of mass over the optimized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

3.2 Spring Flamingo Walking Gait

To analyze a more realistic system, we tested our methods on a planar simulation of the Spring Flamingo robot[13]. 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 transport. Cost of transport is a good, unitless indication of the energy consumption required for locomotion. Where d is the total distance traveled, we write the cost as:

$$g(x, u) = \frac{1}{mgd} \sum_{k=1}^N \sum_i |\dot{q}_{k,i} u_{k,i}| \quad (24)$$

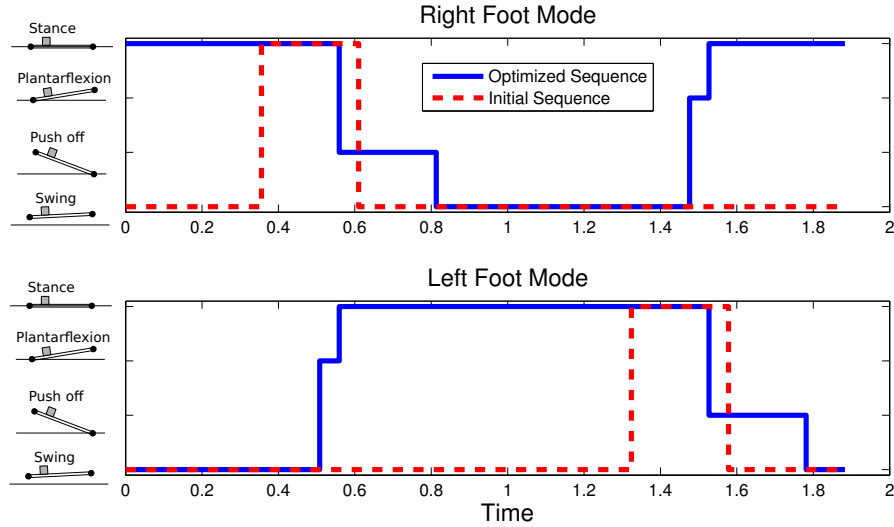


Fig. 6 The optimized mode sequence of the left and right feet is plotted against time and the mode transitions are labeled. The SQP was initialized with the significantly different sequence, demonstrating the ability of the algorithm to independently plan through contact discontinuities. Note that, in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

Since the solutions to the NP 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 λ tape implicitly identifies the 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 tape leads the trajectory to a right-left walking gait but details such as independent heel strike and heel off were identified in the optimization process. To optimize for a cyclic gait, it is natural to write the periodicity constraint:

$$x_N = x_1 \quad (25)$$

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 joints, respectively, and d_{min} is the minimum stride length, we have:

$$\begin{bmatrix} y_N \\ \theta_N \\ \dot{x}_N \\ \dot{y}_N \\ \dot{\theta}_N \\ q_{l,N} \\ q_{r,N} \\ \dot{q}_{l,N} \\ \dot{q}_{r,N} \end{bmatrix} = \begin{bmatrix} y_1 \\ \theta_1 \\ \dot{x}_1 \\ \dot{y}_1 \\ \dot{\theta}_1 \\ q_{r,1} \\ q_{l,1} \\ \dot{q}_{r,1} \\ \dot{q}_{l,1} \end{bmatrix} \quad (26)$$

$$x_N \geq x_1 + d_{min} \quad (27)$$

With these linear constraints and given a nominal trajectory from Pratt’s original work on the robot where the cost of transport was 0.18[12], our methods identified a periodic walking gait which reduced the cost to 0.04. This is a significant reduction in cost and is impressive in its own right, especially 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.

3.3 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.

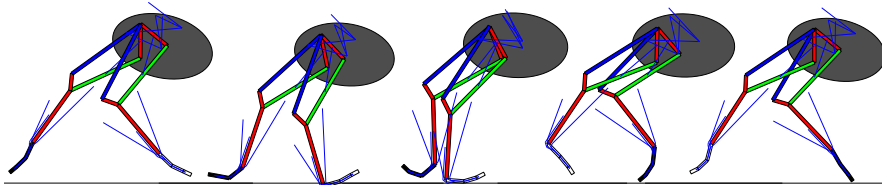


Fig. 7 A generated trajectory for the FastRunner robot running at over 20 mph. The solid elements show the leg linkages and the blue lines indicate springs and tendons. Only the hip joints of the robot are actuated.

Figure 7 shows a motion sequence of an optimized periodic running gait, averaging over 20 mph. As with Spring Flamingo, constraints (26) and (27) 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. 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 NP 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.

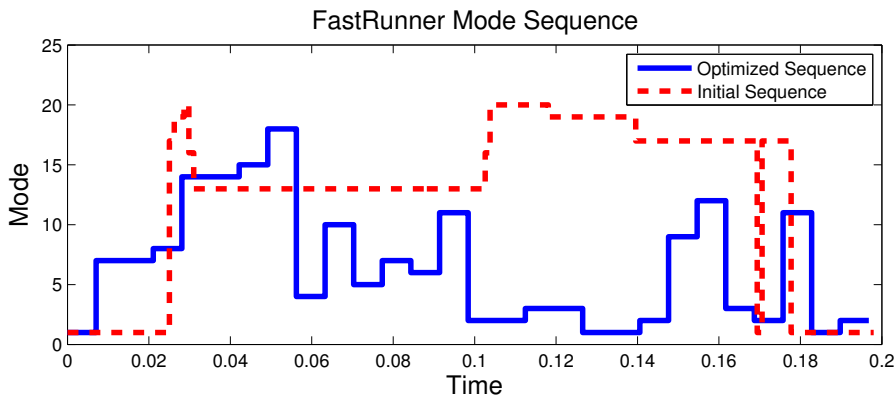


Fig. 8 Out of more than 4 million possible discrete modes, the sequence for one locally optimal cyclic trajectory is shown. This sequence, which passes through 15 different modes, is compressed and plotted above against the largely distinct mode sequence of the initial trajectory.

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 NCP constraint formulation, our methods are now able to generate a locally optimal gait for FastRunner.

4 Discussion and Future work

In our approach, we write a complex NP that in practice has been tractable for state of the art solvers. Since the problem is non-convex, we are limited to locally op-

timal 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. However, we have also found that certain relaxations of the complementarity conditions can greatly improve the rate of convergence and reduce the likelihood of a poor, local solution. In particular, the simple relaxation $\phi(q)\lambda \leq \alpha$ works well in practice. This has the intuitive effect of smoothing out the complementarity conditions by allowing force to be applied at a distance during early phases of the algorithm. This additionally grows the feasible state space of the NP to include regions where the complementarity gradient is more useful to the solver. As α is driven to 0 over a few passes, the slack in the constraint helps allow the solver to converge to a better trajectory by improving the conditioning of the SQP. Many smoothing functions for NPs exist and have been used to directly solve these problems, such as the Fischer-Burmeister function [6] or the class of functions suggested in [4], and these functions may be applicable here as well.

As is mentioned above, throughout this paper we have solely dealt with the case of the robot interacting with a single object in the environment. While this is not a hard constraint, an increase in the number of interacting bodies will lead to a potentially combinatorial number of contacts and corresponding contact forces. Of course, this also implies $O(\exp(n^2))$ potential hybrid modes. We have also focused on purely inelastic collisions where the effective coefficient of restitution vanishes. The works in [1, 11] have 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). We believe that, given a nominal trajectory, this approach will solve the problem of real-time local control as a convex optimization program and that the controller will be capable of planning a finite-horizon path with a different mode sequence than that of the nominal trajectory.

5 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 reference to explicit 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 three 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.

6 Acknowledgments

This work was supported by the DARPA Maximum Mobility and Manipulation program, BAA-10-65-M3-FP-024.

References

1. 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.
2. John T. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming*. SIAM Advances in Design and Control. Society for Industrial and Applied Mathematics, 2001.
3. 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.
4. 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.
5. S. Cotton, I. Olaru, 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.
6. A. Fischer. A special newton-type optimization method. *Optimization*, 24(3-4):269–284, 1992.
7. 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.
8. Philip E. Gill, Walter Murray, and Michael A. Saunders. *User's Guide for SNOPT Version 7: Software for Large -Scale Nonlinear Programming*, February 12 2006.
9. C. R. Hargraves and S. W. Paris. Direct trajectory optimization using nonlinear programming and collocation. *J Guidance*, 10(4):338–342, July-August 1987.
10. 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.
11. 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.
12. 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.
13. 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.

14. G. Schultz and K. Mombaur. Modeling and optimal control of human-like running. *IEEE/ASME Transactions on Mechatronics*, 15(5):783–792, Oct. 2010.
15. 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.
16. Manoj Srinivasan and Andy Ruina. Computer optimization of a minimal biped model discovers walking and running. *Nature*, 439:72–75, January 5 2006.
17. 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.
18. Y. Tassa and E. Todorov. Stochastic complementarity for local control of discontinuous dynamics. In *Proceedings of Robotics: Science and Systems (RSS)*. Citeseer, 2010.
19. K. Wampler and Z. Popovic. Optimal gait and form for animal locomotion. In *ACM Transactions on Graphics (TOG)*, volume 28, page 60. ACM, 2009.
20. 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.