2016 IEEE International Conference on Robotics and Automation (ICRA)
Stockholm, Sweden, May 16-21, 2016

Optimization and stabilization of trajectories for constrained dynamical systems

Michael Posa[1], Scott Kuindersma[2], and Russ Tedrake[1]

***Abstract*— Contact constraints, such as those between a foot and the ground or a hand and an object, are inherent in many robotic tasks. These constraints define a manifold of feasible states; while well understood mathematically, they pose numerical challenges to many algorithms for planning and controlling whole-body dynamic motions. In this paper, we present an approach to the synthesis and stabilization of complex trajectories for both fully-actuated and underactuated robots subject to contact constraints. We introduce a trajectory optimization algorithm (DIRCON) that extends the direct collocation method, naturally incorporating manifold constraints to produce a nominal trajectory with third-order integration accuracy—a critical feature for achieving reliable tracking control. We adapt the classical time-varying linear quadratic regulator to produce a local cost-to-go in the manifold tangent plane. Finally, we descend the cost-to-go using a quadratic program that incorporates unilateral friction and torque constraints. This approach is demonstrated on three complex walking and climbing locomotion examples in simulation.**

I. Introduction

Many of the fundamental problems in robotics, such as locomotion and manipulation, involve the robot making and breaking contact with its environment. The last few years have seen rapid advances in motion planning techniques, based on trajectory optimization, to synthesize locally optimal trajectories which make and break contact, even for very complex robots and tasks [22], [24], [29], [33]. However, there has been relatively little work on stabilizing the resulting plans using local feedback for robust execution, with [29] a notable exception. Our own initial attempts to stabilize these trajectories were thwarted by numerical difficulties related to insufficient accuracy in the motion plans.

Contact, when sustained over time, represents kinematic constraints on the evolution of the dynamical system. In simple cases, the constrained system can be described by a set of minimal coordinates. However, these constraints frequently create closed kinematic chains, such as in four-bar linkages, or when a walking robot is in double support with both legs contacting the ground. When minimal coordinates do not exist, we must consider the robot's state to be on a manifold embedded in a higher dimensional space [21], [4].

In this paper, we provide extensions to the widely used direct collocation trajectory optimization algorithm [13] and classical linear quadratic regulator (LQR) stabilization technique that address the challenges posed by working on these manifolds. We introduce DIRCON, a planning algorithm for constrained dynamical systems with third-order integration accuracy; it maintains the advantages of direct collocation, thereby enabling more reliable stabilization as compared with existing methods. The results are then combined with recent advances in humanoid control, leveraging quadratic programming, to incorporate constraints such as input saturations and friction limits into the feedback policy [19], [18]. These three elements provide an end-to-end recipe for generating and stabilizing optimal trajectories that exhibit complex and varying contact configurations. We demonstrate the approach on three different locomotion examples in simulation: walking in three dimensions, underactuated planar (2D) walking, and planar climbing utilizing contact between the hand and the environment.

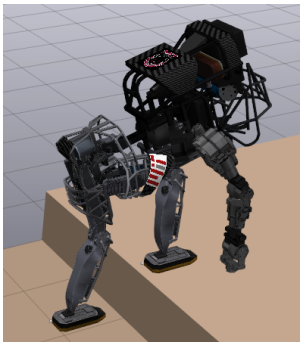

Fig. 1. An Atlas robot climbing a step with aid of a hand. The algorithms presented here synthesize and stabilize dynamic, multi-contact trajectories.

II. Background and Related Work

There is an extensive literature related to planning and controlling rigid-body systems through intermittent contacts. In the following section, we summarize the background material and related work necessary to motivate our contributions.

A. Constrained Dynamics

In this paper, we will consider constrained Lagrangian systems [10], [21], which we briefly review here. Letting $q, v \in \mathbb{R}^n$ be the generalized position and velocity coordinates, we have the state vector, $x = \begin{bmatrix} q \\ v \end{bmatrix}$. Take the control input $u \in \mathbb{R}^m$ and consider dynamics of the standard form:

$$\dot{q}(t) = v(t), \quad \dot{v}(t) = f(x(t), u(t)), \tag{1}$$

such that the dynamics constrain the system evolution to always satisfy the d-dimensional constraint $\phi(q(t)) = 0$ for $\phi : \mathbb{R}^n \to \mathbb{R}^d$. With ϕ a smooth mapping, this constraint has the effect of restricting trajectories to a $(2n-2d)$-dimensional

[1]CSAIL, Massachusetts Institute of Technology, Cambridge, MA, USA. {mposa,russt}@csail.mit.edu
[2]School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA. scottk@seas.harvard.edu
This work supported by AFRL Award No. FA8750-12-1-0321 and NSF Award No. 724454.

manifold. For the purposes of trajectory optimization, we will also write the dynamics in an implicit form:

$$\dot{v}(t) = \bar{f}(x(t), u(t), \lambda(t)), \tag{2}$$

with the constraint force $\lambda \in \mathbb{R}^d$. Since the constraint must be satisfied over the entirety of a trajectory, $x(t)$, the time derivatives of $\phi(q(t))$ must also vanish. Defining the Jacobian $J(q) = \dfrac{\partial \phi}{\partial q}$, we know that trajectories and constraint forces must satisfy the following conditions:

$$\phi(q(t)) = 0, \tag{3}$$

$$\psi(q, v) \equiv \frac{\mathrm{d}\phi}{\mathrm{d}t} = J(q)v = 0, \tag{4}$$

$$\alpha(q, v, u, \lambda) \equiv \frac{\mathrm{d}^2\phi}{\mathrm{d}t^2} = \frac{\mathrm{d}J(q)}{\mathrm{d}t}v + J(q)\bar{f}(x, u, \lambda) = 0, \tag{5}$$

where we have defined $\psi(q, v)$ and $\alpha(q, v, u, \lambda)$ as the constraint velocity and acceleration, respectively.

We assume that, for given initial conditions, there exists a unique solution $x(t)$ to the constrained dynamics. Since $J(q)$ may have a non-empty nullspace, we do *not* require that $\lambda(t)$ be unique. We assume that $J(q)$ has constant rank. Let $\lambda^*(q, v)$ define a constraint force that satisfies the acceleration constraint, $\alpha(q, v, u, \lambda^*(q, v)) = 0$ for all q and v. In Section III, we will present strategies for trajectory optimization and control of constrained dynamical systems.

B. Trajectory optimization

There is a rich literature on both control and planning of nonlinear systems as applied to mobile robotics. Trajectory optimization has been particularly successful in synthesizing highly dynamic motions in high-dimensional state spaces. See Betts [3] for both an overview and a description of the variety of existing algorithms. Broadly speaking, trajectory optimization aims to find dynamically consistent state and control trajectories $x(t), u(t)$ that minimize a cost functional subject to a set of constraints. A popular class of techniques, commonly known as transcription methods, discretizes the trajectories in time as $x_1, ..., x_N, u_1, ..., u_N$ and, between these knot points, enforces the integral of the dynamics as a constraint. Within this class are multiple-shooting methods, which numerically integrate from x_k to x_{k+1} and have been successfully applied to robotic locomotion tasks [22], [27]. Another common approach, called direct collocation, avoids costly numerical integration by approximating trajectories as Hermitian splines [13] (Section II-C). Remy [26] used direct collocation with pre-specified contact sequences to generate 1-leg hopping and 2-leg gaits. Buss et al. [5] used direct collocation in minimal coordinates to optimize walking phases for a medium-scale humanoid robot.

When applied to a constrained dynamical system, it is natural to require that the points x_k lie on the constraint manifold and satisfy (3)-(4). This poses a challenge due to inherent error in numerical integration methods; naive application of standard trajectory optimization methods will be unable to satisfy these additional constraints. However, corrective techniques for integration on manifolds exist, such as augmenting the dynamics to force trajectories to drift back toward the manifold [6], [11].

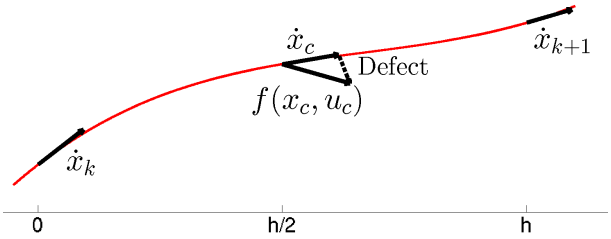

Fig. 2. The Hermite spline of the direct collocation algorithm, shown in red, is constructed between two knot points. The defect is the discrepancy between the collocation dynamics $f(x_c, u_c)$ and the slope of the spline $\dot{x}_c$.

Recent research has seen the development of contact-implicit trajectory optimization algorithms, capable of synthesizing motions without an *a priori* specification of the contact sequence [24], [23]. Based on approaches widely used in simulation, a first-order integration method was used in [24]. However, the integration error induced by such a low-accuracy method greatly complicates the tasks of trajectory execution and stabilization, an effect also observed in [33]. Here, we will assume that the contact sequence is specified, which in practice can be provided by a contact-implicit method or by the designer. In Section III-A, we introduce a third-order integration method, motivated by classical Hermite-Simpson integration, that does not require minimal coordinates and is especially adapted to the task of optimal control on manifolds.

C. Direct Collocation

The original direct collocation algorithm, introduced by Hargraves and Paris, uses cubic Hermite splines to interpolate between a sequence of knot points [13]. The state and input are defined at a sequence of equally spaced knot points at times $t_1, .., .t_N$ such that the timestep between sequential points is h. This defines the list of decision parameters $z = (x_1, ..., x_N, u_1, ..., u_n)$. We briefly describe the algorithm here, as applied to a second-order system.

The plant dynamics are evaluated at each knot point $\dot{v}_k = f(x_k, u_k)$ and, for every sequential pair of knot points x_k, x_{k+1} and inputs u_k, u_{k+1}, a cubic spline $x_s : [t_k, t_k + h] \to \mathbb{R}^{2n}$ is generated that matches the state and its first derivative at the knot points:

$$x_s(t_k) = x_k, \qquad x_s(t_{k+1}) = x_{k+1},$$

$$\dot{x}_s(t_k) = \begin{bmatrix} v_k \\ f(x_k, u_k) \end{bmatrix}, \qquad \dot{x}_s(t_{k+1}) = \begin{bmatrix} v_{k+1} \\ f(x_{k+1}, u_{k+1}) \end{bmatrix}.$$

The state at the midpoint of this spline, $x_c = x_s(t_k + .5h)$, called the collocation point, is simply a linear combination of the state and its derivative at the adjacent knot points. The collocation constraint function, g, matches the slope of the spline to the plant dynamics at the collocation point, where the control input is typically taken to be the result of a first-order hold, $u_c = \frac{u_k + u_{k+1}}{2}$. Therefore, we define:

$$g(x_k, u_k, x_{k+1}, u_{k+1}) = \dot{x}_s(t_k + .5h) - \begin{bmatrix} v_c \\ f(x_c, u_c) \end{bmatrix}. \tag{6}$$

These collocation constraints, illustrated in Figure 2, are an efficient and accurate representative of the plant dynamics. The integration error over a single timestep is $\mathcal{O}(h^4)$ and so the error over a fixed time interval is then $\mathcal{O}(h^3)$ [12]. This third-order accuracy compares favorably with Euler-integration based methods, which have only $\mathcal{O}(h)$ accuracy over similar intervals. With higher order methods, accurate trajectories can be achieved with fewer knot points–and therefore smaller optimization problems. The resulting trajectory optimization problem, with running cost $\ell(x_k, u_k)$ and a final cost $\ell_f(x_N)$ can then be expressed as:

$$\begin{aligned} \underset{z}{\text{minimize}} \quad & \ell_f(x_N) + h\sum_{k=1}^{N}\ell(x_k, u_k) \\ \text{subject to} \quad & 0 = g(x_k, u_k, x_{k+1}, u_{k+1}) \\ & \quad \text{for } k = 1, \ldots, N-1 \\ & 0 \geq m(z), \end{aligned} \tag{7}$$

where $m(z)$ represents arbitrary additional constraints on state and input, such as variable bounds or boundary values.

D. Control of walking systems

The most widespread algorithms applied to modern robots are those that optimize and track trajectories using a reduced dynamical model. The power of these approaches was demonstrated at the DARPA Robotics Challenge in June 2015, where highly complex robots were able to move through a challenging course with little hesitation on the part of their algorithms (but considerable hesitation on the part of their operators). There is a large literature on trajectory design and stabilization for legged robots, using dynamic quantities like the zero moment point [15] or the capture point [16]. The simple expressions for these quantities admit efficient algorithms for planning and computing optimal controllers [31] and can form the basis for tracking controllers that utilize Quadratic Programs (QPs) to reason, real-time, about the constraints of the system [19], [17], [8], [14]. Although these approaches are versatile and computationally simple, they fall short as tools for generating highly dynamic and energetically-efficient motions in underactuated systems.

优化控制问题根据L和g的分类变成了LQR

Dai et al. [7] showed that by starting with the full coordinates and removing torque limits, direct trajectory optimization problems can be formulated considering only the floating base, or Centroidal, dynamics and the full body kinematics. This leads to a more tractable nonlinear optimization problem but is restricted to fully-actuated robots and removes the ability to reason about torque limits and costs. Tassa et al. [29] use smoothed contact models to achieve short-horizon motion planning through contact at online rates using differential dynamic programming.

There are also successful examples of dynamic walking systems that do not use trajectory optimization. Westervelt et al. [32] developed the hybrid zero dynamics (HZD) framework whereby virtual holonomic constraints are defined and tracked in the minimal coordinates. This approach has been used to produce dynamic walking and running examples in physical robots [28]. Ames et al. [1] extended this line of work by using QPs to formulate exponentially stabilizing control-Lyapunov controllers for HZD-based walking systems. The challenge for HZD approaches remains to extend to more general, aperiodic motions.

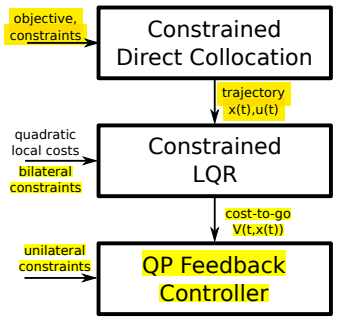

Fig. 3. A block diagram shows the interaction between the three components. The DIRCON algorithm, given an objective and constraints, produces a nominal trajectory. The constrained version of LQR solves for a quadratic cost-to-go function. This cost-to-go is used to synthesize a QP, which is solved in real-time as a feedback control policy.

Our current approach differs from previous work in that we optimize trajectories for constrained systems in the full coordinates with realistic impact models. The numerical accuracy of these trajectories is sufficient to stabilize using a combination of techniques that can be considered as standard: LQR and QP.

III. Approach

Planning and control is achieved via three components, detailed in the following sections and illustrated in Figure 3. The direct collocation algorithm is extended to seamlessly produce trajectories for constrained dynamical systems that also obey Coulomb friction limits. An extension of the LQR synthesizes a local controller and cost-to-go function, valid near the nominal motion. Rather than directly applying the LQR controller, the cost-to-go is used as the objective of a QP that is solved, real-time, as a feedback policy that respects inequalities and changing contact conditions.

A. Constrained Direct Collocation

Observe that simply adding in the manifold constraints (4)-(5) to the standard direct collocation optimization (7) results in an over-constrained problem. This can best be seen by formulating the optimization as a single-step forward prediction: fix x_0, u_0, and u_1 and solve for x_1. Assuming that x_0 lies on the manifold, we still have the constraints (4), (5), and (6), a total of $(2n+2d)$ equality conditions, greater than the dimensionality of the unknown x_1.

To resolve this issue, we present Constrained Direct Collocation (DIRCON), an extension of the classical algorithm that naturally handles the difficulties presented by an implicit constraint manifold. This algorithm has two main contributions: 1) it achieves $\mathcal{O}(h^3)$ accuracy for constrained Lagrangian systems and 2) by *explicitly* representing the forces λ, constraints on the forces like friction limits are easily expressed. First, the algorithm incorporates the constraint forces at the knot points, $\lambda_1, \ldots, \lambda_N$, as explicit decision variables in the optimization. Second, it reduces the effective

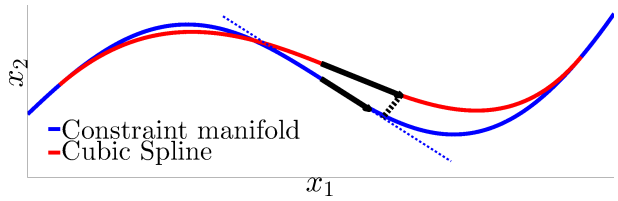

Fig. 4. A one dimensional constraint manifold, embedded in a two dimensional space, is cartooned in blue. A Hermite spline, in red, between two points will not overlap the manifold, and its slope will not lie within the tangent plane at the collocation point. DIRCON implicitly projects the spline slope onto the manifold to form the proper constraint defect.

dimensionality of (6) by restricting it to the tangent plane of the constraint manifold, through the use of additional slack variables $\bar{\lambda}_1, ..., \bar{\lambda}_{N-1}, \bar{\gamma}_1, ..., \bar{\gamma}_{N-1}$. These variables represent forces and a velocity correction, respectively, applied at the collocation point. The resulting projected collocation constraint, cartooned in Figure 4, is:

$$\bar{g}(x_k, u_k, \lambda_k, x_{k+1}, u_{k+1}, \lambda_{k+1}, \bar{\lambda}_k, \bar{\gamma}_k) = \dots$$
$$\dot{x}_s(t_k + .5h) - \begin{bmatrix} v_c + J(q_c)^T \bar{\gamma}_k \\ \bar{f}(x_c, u_c, \bar{\lambda}_k) \end{bmatrix}. \quad (8)$$

Defining the set of optimization parameters as $z = (x_1, ..., x_N, u_1, ..., u_N, \lambda_1, ..., \lambda_N, \bar{\lambda}_1, ..., \bar{\lambda}_{N-1}, \bar{\gamma}_1, ..., \bar{\gamma}_{N-1})$, we have the trajectory optimization problem:

$$\begin{aligned} \underset{z}{\text{minimize}} \quad & \ell_f(x_N) + h\sum_{k=1}^{N} \ell(x_k, u_k) \\ \text{subject to} \quad & 0 = \bar{g}(x_k, u_k, \lambda_k, x_{k+1}, u_{k+1}, \lambda_{k+1}, \bar{\lambda}_k, \bar{\gamma}_k) \\ & \quad \text{for } k = 1, ..., N-1 \\ & 0 = \phi(q_k) = \psi(x_k) = \alpha(q_k, v_k, u_k, \lambda_k) \\ & \quad \text{for } k = 1, ..., N \\ & 0 \geq m(z). \end{aligned} \quad (9)$$

As with standard collocation, additional constraints on the state, input, and constraint forces can all be added to supplement the trajectory optimization.

Theorem 1: If the dynamics and kinematics functions $(\bar{f}, \lambda^*, \phi, \psi, \alpha)$ are analytic and Lipschitz continuous, the algorithm above has $\mathcal{O}(h^3)$ accuracy over a fixed time-interval. More specifically, take $(x_0, u_0, \lambda_0, x_1, u_1, \lambda_1, \bar{\lambda}_0, \bar{\gamma}_0)$ to be bounded and satisfy (8). Let $x(t)$ for $t \in [0, h]$ be the true solution to $\dot{x}(t) = f(x(t), u(t))$ with $x(0) = x_0$ and $u(t)$ a first-order hold between u_0 and u_1. Then, we have that the error $||x(h) - x_1|| < Ch^4$ for some constant C.

Proof: Let $q_s(t)$ and $v_s(t)$ correspond to the joint position and joint velocity cubic splines. For simplicity, we will write $q_c = q_s(.5h)$, $\dot{q}_c = \dot{q}_s(.5h)$, $v_c = v_s(.5h)$, and $\dot{v}_c = \dot{v}_s(.5h)$. Note, because the parameters z are bounded, $q_s(t)$ and $v_s(t)$ are also bounded and so $\phi(q_s(t))$ will be both bounded and analytic. First, we demonstrate that $\phi(q_c) = \mathcal{O}(h^4)$. Since $\phi(q_s(0))$ and $\dot{\phi}(q_s(0))$ both vanish, the Taylor expansion of $\phi(q_s(t))$ is

$$\frac{t^2}{2}\frac{\mathrm{d}^2\phi}{\mathrm{d}t^2}\bigg|_{t=0} + \frac{t^3}{6}\frac{\mathrm{d}^3\phi}{\mathrm{d}t^3}\bigg|_{t=0} + \frac{t^4}{24}\frac{\mathrm{d}^4\phi}{\mathrm{d}t^4}\bigg|_{t=0} + \dots$$

By substituting and differentiating this expansion, and exploiting the fact that $\phi(q_s(h))$ and $\dot{\phi}(q_s(h))$ also vanish, we can eliminate the quadratic and cubic terms from $\phi(q_c)$,

$$\phi(q_c) = \frac{233}{1142}h^4 \frac{\mathrm{d}^4\phi}{\mathrm{d}t^4}\bigg|_{t=0} + \mathcal{O}(h^5). \quad (10)$$

Therefore, for sufficiently small h we have $||\phi(q_c)|| < Ch^4$ and, similarly, $||\dot{\phi}(q_c)|| < Ch^3$. For notational ease, we take C to be some global constant of sufficient size. A similar expansion of $\psi(q_s(t), v_s(t))$ demonstrates that $||\psi(q_c, v_c)|| < Ch^4$ and $||\dot{\psi}(q_c, v_c)|| < Ch^3$. The bounds on these two values for the constraint velocity, $\dot{\phi}(q_c)$ and $\psi(q_c, v_c)$, combined with the collocation constraint (8) give a bound on the velocity correction,

$$||J(q_c)^T \bar{\gamma}_0|| < Ch^3. \quad (11)$$

Combined with the velocity component of (8), we have

$$||\alpha(q_c, v_c, u_c, \bar{\lambda}_0)|| < Ch^3. \quad (12)$$

Simply put, (11) and (12) bound the defect between the derivative of the splines and the manifold tangent plane. To leverage existing results regarding collocation methods and ODEs, we extend the constrained dynamics by defining $\dot{x}$ when x is off the manifold,

$$\dot{q}(t) = v(t) + J(q(t))^T \gamma(t) \quad (13)$$
$$\dot{v}(t) = f(x(t), u(t), \lambda(t)), \quad (14)$$

where the constraint forces are such that $J(q(t))\dot{q}(t) = 0$ and $\alpha(q(t), v(t), u(t), \lambda(t)) = 0$. Note that these extended dynamics agree with the constrained dynamics for states on the manifold, but define an ODE for all $x \in \mathbb{R}^n$. Using standard results in Hairer [12] and Betts [3], a direct collocation algorithm for these extended dynamics would have $\mathcal{O}(h^4)$ accuracy over a single timestep. While (11) and (12) imply $\mathcal{O}(h^3)$ errors in extended dynamics when evaluated at the collocation point, this error is multiplied by h in computation of the integral and so the overall accuracy is still $\mathcal{O}(h^4)$. ■

1) Friction Limits: By explicitly introducing the constraint forces λ as decision parameters within the optimization, we can easily require that they obey a set of nonlinear constraints as in [24]. For instance, we can require that they lie within the Coulomb friction cone $\mu^2\lambda_z^2 \geq \lambda_x^2 + \lambda_y^2$, where λ_z is the component normal to the contact surface.

This is of particular interest when the rows of the Jacobian J are *not* linearly independent, a common case that occurs in all of the examples in this paper. When J is full row-rank, one might directly solve for the unique λ such that $\alpha(q, v, u, \lambda) = 0$ and evaluate the friction constraints by solving a simple linear system of equations. However, when J is rank deficient, there are a subspace of such forces. Therefore, solving for a force that satisfies the constraints is equivalent to a convex optimization problem in and of itself. Explicit representation of the forces avoids this added complexity, and greatly simplifies the representation of these constraints.

2) Hybrid Collocation: As with other trajectory optimization algorithms, DIRCON can be simply extended to the hybrid case. The hybrid trajectory optimization problem constructs one set of decision parameters and constraints per contact state, or hybrid mode. Consistency between the modes is enforced via a hybrid jump condition, described by explicitly including the impulse Λ. Letting z^j be the decision variables for the jth mode, jump constraints are then:

$$q_1^j = q_{N^{j-1}}^{j-1} \tag{15}$$
$$v_1^j = v_{N^{j-1}}^{j-1} + G(q_{N^{j-1}}^{j-1}, \Lambda^{j-1}), \tag{16}$$

where $G(q, \Lambda)$ represents the change in velocity that results from applying impulse Λ at the active contact points in mode j at the given position q. Additional constraints prevent contact penetration, and guard conditions to ensure that mode changes occur when the appropriate points are in contact.

B. Equality-Constrained LQR

Given a trajectory output from the collocation algorithm described in the previous section, we next address the problem of designing a tracking controller. The presentation here is similar in principle to [21], though we base the design around LQR. A powerful tool for the stabilization of both time-invariant and time-varying linear dynamical systems, LQR is also widely used for *local* stabilization of nonlinear systems [2]. For a linear system, finite horizon LQR minimizes the quadratic cost,

$$x(T)^T Q_f x(T) + \int_0^T [x(t)^T Q x(t) + u(t)^T R u(t)] \mathrm{d}t, \tag{17}$$

by solving the Hamilton-Jacobi-Bellman (HJB) equation. The product is an optimal controller $u(t) = -K(t)x(t)$ and the cost-to-go $V(t, x(t)) = x(t)^T S(t) x(t)$. To track a trajectory of a nonlinear system, a linearization of the dynamics about the nominal motion can be used to generate a feedback policy. Here, we provide a straight-forward extension of the classical notion of LQR to constrained dynamical systems. Consider the time-varying linear system

$$\dot{x} = A(t)x(t) + B(t)u(t), \tag{18}$$

where the dynamics constrain the state to the manifold defined by $F(t)x(t) = 0$ and $F(t)$ is full row-rank. While the derivations in this section apply to generic systems, for notational consistency, we will continue to focus on second-order plants with $F : \mathbb{R}^+ \to \mathbb{R}^{(2n-2d)\times 2n}$. The manifold constraint implies that the system is neither controllable nor stabilizable in the traditional senses. As a result, we cannot simply ignore $F(t)$ and solve the standard Riccati equation.

While we may not have a set of minimal coordinates, we can derive a time-varying basis for locally minimal coordinates and then apply traditional LQR techniques. Assume that $F(t)$ is differentiable and take some $P(0)$ to be an orthonormal basis of the nullspace of $F(0)$:

$$PP^T = I^{2d} \tag{19}$$
$$PF^T = 0^{2d\times(2n-2d)}. \tag{20}$$

To ensure that these identities hold for all time, we differentiate and write an ordinary differential equation for $P(t)$,

$$\begin{aligned} \dot{P}P^T + P\dot{P}^T &= 0 \\ \dot{P}F^T + P\dot{F}^T &= 0. \end{aligned} \tag{21}$$

For any $x \in \mathbb{R}^{2n}$, we can write $x = P^T y + F^T z$ for some $y \in \mathbb{R}^d$ and $z \in \mathbb{R}^{n-d}$. However, as a result of the constraints, we know that $Fx(0) = z(0) = 0$. Additionally, along any trajectory $x(t)$, we have $\dot{z}(t) = 0$ and so $x(t) = P^T y(t)$ and $y(t) = Px(t)$. The dynamics of y are given by:

$$\dot{y} = \dot{P}x + P\dot{x} = \bar{A}y + \bar{B}u, \tag{22}$$

for $\bar{A} = \dot{P}P^T + PAP^T$ and $\bar{B} = PB$. We can apply classical LQR control techniques to this system in a two-step process. First, given $F(t)$, generate an appropriate $P(0)$ and then numerically integrate (21) to find $P(t)$. Note that some regularization of the ODE may be required to ensure that the solution does not drift from the identities (19)-(20). Second, use $P(t)$ to perform the change of coordinates from x to y. Solve the resulting Riccati equation and transform the solution back to the original coordinates.

The LQR solution from an individual mode can be projected through hybrid transitions using a linearization of the instantaneous impact dynamics, via the jump Riccati equation described in [20].

1) Example: Kinematic Constraint: We cast the case of a kinematic constraint $\phi(q)$ into the constrained LQR formulation. Given a nominal trajectory $q_0(t), v_0(t)$ that satisfies the constraints $\phi(q_0(t)) = 0$ and $\psi(q_0(t), v_0(t)) = 0$, linearize the dynamics about this trajectory:

$$\begin{aligned} q(t) &= q_0(t) + \tilde{q}(t), \quad v(t) = v_0(t) + \tilde{v}(t) \\ \dot{\tilde{q}}(t) &= \tilde{v}(t) \\ \dot{\tilde{v}}(t) &= \tilde{A}(t) \begin{bmatrix} \tilde{q}(t) \\ \tilde{v}(t) \end{bmatrix}. \end{aligned}$$

Linearizing the constraint, and suppressing the dependence of q_0 and v_0 on time, we get

$$\begin{bmatrix} \phi(q) \\ \psi(q, v) \end{bmatrix} \approx \begin{bmatrix} J(q_0) & 0 \\ \dfrac{\mathrm{d}J(q_0)}{\mathrm{d}t} & J(q_0) \end{bmatrix} \begin{bmatrix} \tilde{q}(t) \\ \tilde{v}(t) \end{bmatrix} = F(t) \begin{bmatrix} \tilde{q}(t) \\ \tilde{v}(t) \end{bmatrix} \tag{23}$$

which gives $F(t)$ as a kinematic function of nominal trajectory. Since we require $F(t)$ to be full rank, but $J(q)$ will often be rank deficient (though constant rank), it is necessary to extract a full rank basis for J and its time derivative.

C. QP Feedback Controller

Rather than executing the time-varying linear policy output from the constrained LQR algorithm, we instead solve a constrained minimization at each control step. This allows us to explicitly take input and friction limits into account and handle minor variations in the timing of impacts. The optimization problem takes the form of a QP. Given a

planned nominal trajectory $x_0(t), u_0(t)$ and LQR solution, we formulate:

$$\begin{aligned} \underset{u,\beta}{\text{minimize}} \quad & \tilde{u}^T R \tilde{u} + 2\tilde{x}^T S(A\tilde{x} + B\tilde{u}) \\ \text{subject to} \quad & H\ddot{q} + C = Bu + J_\beta^T \beta \\ & \dot{J}\dot{q} + J\ddot{q} = 0 \\ & u_{\min} \leq u \leq u_{\max} \\ & \beta \geq 0, \end{aligned} \tag{24}$$

where $\tilde{x} = x - x_0(t)$, $\tilde{u} = u - u_0(t)$. We have suppressed dependence on time and state. The cost function is derived from the HJB equation for the time-varying LQR system. Therefore, in the absence of unilateral constraints, the QP solution is equivalent to the optimal LQR input. The Riccati matrix S is evaluated based on time, with the exception that if an impact occurs early, within a few timesteps of the planned impact, S is evaluated from the next mode. This helps avoid large errors in velocity states at impacts.

The decision variables β are force coefficients that multiply a set of generating vectors that define a polyhedral approximation to the friction cone, $\lambda_j = \sum_{i=1}^{N_d} \beta_{ij} w_{ij}$, where $w_{ij} = n_j + \mu_j d_{ij}$, n_j, d_{ij} are the contact-surface normal and i^{th} tangent vector for the j^{th} contact point, respectively, μ_j is the friction coefficient, and N_d is the number of tangent vectors used in the approximation. The contact points included in β are determined at each control step. Note that any point in contact will be added to the QP, whether or not it is planned, giving the system the opportunity to use environmental forces to correct deviations from the desired trajectory.

The second equation in (24) acts as a "no slip" constraint by requiring that the planned contact points do not accelerate with respect to the world frame when they are active. The Jacobian J, as previously defined, maps joint velocities to Cartesian velocities of the contact points. J_β^T represents the use of the generating vectors, and so maps forces along the generating vectors into generalized forces. In practice, we often soften the constraint $\dot{J}\dot{q} + J\ddot{q} = \eta$, where the slack variable η is penalized quadratically. This formulation shares several features with our QP-based controller used on Atlas [19], [18], with the important distinction that the local cost-to-go is in the full coordinates.

IV. Experiments

The components above are tested on three examples related to robotic locomotion. The algorithms were implemented in MATLAB within the Drake planning and control toolbox [30], also used for simulation, and the source code is all openly available online[1]. Trajectory optimizations were solved with the SNOPT toolbox [9]. Depending on complexity, the trajectory optimization and LQR components were solved offline on a desktop computer within ten minutes to two hours. The QP controller was solved at real-time rates during simulation. Highlights from the experiments here are shown in figures below, and the full executions are available within the accompanying video[2].

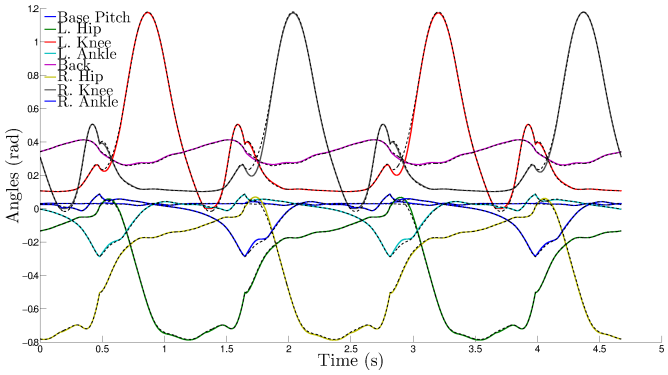

Fig. 5. Joint angle and body pitch tracking error along four steps of the underactuated walker. The nominal trajectory is shown in the dashed lines and the executions are in the solid lines. Tracking error is worst shortly following the impacts (where the trajectories are not differentiable).

A. Underactuated planar biped

We first demonstrate the approach on underactuated planar biped, where each leg has a degree of freedom in the hip, knee, and ankle. The hips and knees are actuated, but the ankle consists solely of a passive spring and damper. With a back joint and the planar floating base, this model has an 20-dimensional state space. Contact points are modeled at the toe and heel of each foot. The mass properties are similar to those of the Atlas robot [18] and the ankle spring and damping coefficients are 10 Nm/rad and 2 Nms/rad. To produce limit cycle walking, a hybrid trajectory optimization was executed with a contact sequence containing both single and double support phases. The objective was to minimize effort, a quadratic penalty on control input $u^T u$. Linear constraints were imposed on x_1 and x_N to produce a periodic motion and a minimum average walking speed. A small penalty was added to acceleration, $10^{-4} \sum_k ||\bar{f}(x_k, u_k, \lambda_k)||^2$, to encourage smoothness in the solution. Additional constraints on the foot position enforced some amount of swing clearance. For the LQR component, simple, diagonal matrices were used for both Q and R. Elements of Q were 100 and 1 for the generalized positions and velocities respectively, while the diagonal R was uniformly 0.01.

Figure 5 shows body pitch and joint angle tracking over four steps. Overall, the controller is able to closely track the nominal motion, with deviations most noticeable shortly after impacts with the ground. One of the aims of this trajectory optimization is to create motions that are both dynamic and efficient. Mechanical cost of transport (COT) serves as a useful metric for locomotion efficiency: if M is the mass and d is the total distance traveled, we integrate the total joint work done and the unitless cost of transport is $\text{COT} = \frac{1}{Mgd} \int |\text{work}| dt$. While we did not explicitly minimize COT, minimizing effort produced an efficient nominal gait with a COT of 0.139. Execution of the trajectory should increase this cost, as the controller must expend energy to eliminate

[1] http://drake.mit.edu and https://github.com/mposa/

[2] http://web.mit.edu/mposa/www/

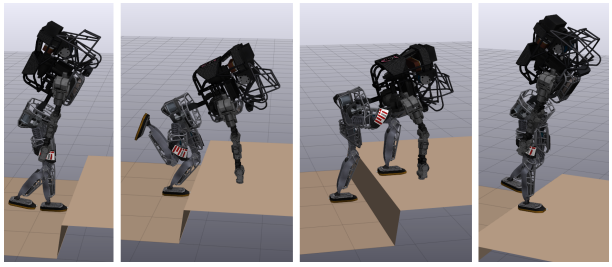

Fig. 6. A sequence of states from the executed trajectory as the robot uses its arm to help climb up onto the step in less than 3 seconds.

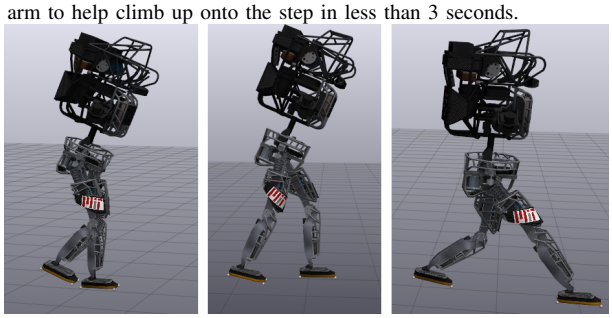

Fig. 7. A sequence of states as the robot walks and then executes a sudden halting maneuver. The leftmost two images illustrate phases of the walking motion and the rightmost image shows the final state after the rapid stop.

error. However, as an indication of the accuracy of the nominal motion, the executed COT over the four steps in the figure was only 0.143–a marginal increase.

B. Multi-contact climbing

We examine a planar humanoid model where the biped from the previous example has been augmented with a two degree of freedom arm (shoulder and elbow joints), also based off the Atlas. Note that the previously used back joint has been eliminated for simplicity and the ankle joints are actuated, giving the plant a 22-dimensional state space. A single contact point is included at the end of the arm, for a total of five possible contacts. The restrictive joint limits of the physical Atlas robot have been relaxed to allow greater flexibility for this motion. The trajectory optimization was constrained to use both the hand and feet to climb a 0.3 meter step and then reach a stable position using a sequence of five different contact modes. As before, constraints were used to enforce swing clearance and the objective was to minimize effort. The costs used for LQR are identical in nature to those from the walking example. The duration of the resulting trajectory was less than 3 seconds, so the robot must move quickly and dynamically to execute it successfully. Figure 6 shows a set of illustrative key-frames from the motion.

C. 3D biped

The final example is a biped walking in three dimensions along flat terrain. Also based upon the Atlas robot, we use a model with six degrees of freedom in each leg: three at the hip, one at the knee, and two at the ankle. Including the floating base, the model has a 36-dimensional state space with eight total contact points at the corners of the feet. As with the underactuated biped, we synthesize limit cycle walking with both single and double support phases. The objective was to minimize effort, and linear constraints enforced a periodic motion while walking at a human-like speed of over 1 m/s. Locomotion at this speed requires continuous, dynamic motion and the planned motion utilizes push-off from the ankle during double support.

To illustrate the capability to produce and execute rapid, aperiodic motions, we further synthesized a trajectory that brought the robot to a complete halt within a half a stride (starting from mid-swing). To stop this quickly, the robot must quickly propel its swing leg forward before coming to rest with its forward foot and rear toe in contact with the ground. For the LQR component, simple, diagonal matrices were again used for both Q and R. Components of Q were 200 and 1.5 for the generalized positions and velocities respectively, while the diagonal R was uniformly 0.01.

We note that, when executed in simulation, the periodic gait was not stable over an infinite horizon as small tracking errors in the footfall locations and timings caused eventual instability. Over shorter distances, however, the controller produces efficient walking at human speeds. The accompanying video demonstrates the robot taking four steps before executing the stopping maneuver, with key-frames shown in Figure 7. As evidence of the accuracy of the nominal trajectory and the efficiency of the feedback policy, the calculated COT of the executed trajectory was 0.399, only slightly larger than the COT of the nominal motion, 0.382.

To examine the response of the closed-loop controller to disturbances, randomly oriented $10N{\cdot}s$ impulses were applied, mid-stance, at the pelvis of the robot and a single walking step was simulated. The impulse causes a roughly 10% deviation in the center of mass velocity and substantial joint velocity errors. The LQR cost-to-go after the disturbance and after one step was used as a measure of robustness. After the disturbance, the median cost-to-go from 300 trials was 0.39 and, after a single step, the controller had reduced the median cost-to-go to 0.07. Some of the random disturbances did cause falls or other instabilities. In total, the controller was able to reduce the cost-to-go in 96% of the trials, empirically demonstrating some level of robustness.

V. Discussion

The most common trajectory optimization approaches utilize tools from general nonlinear optimization and are therefore sensitive to the choice of the initial, or seed, value for the unknown parameters. As an indication of the robustness of the DIRCON method, all of the optimizations in Section IV were initialized with exceedingly simple trajectories: a constant, nominal pose for the states and white noise for the control inputs. Even without carefully chosen seed values, the optimizations consistently converged to high quality solutions for problems of significant size.

While we are able to execute motions over finite horizons, the walking trajectories discussed above are not stable over an infinite sequence of steps. Perturbations around around the moment of impact cause slight mismatches in footfall timing and location, eventually leading to a fall. One potential solution to this issue is to eliminate the explicit dependence

of the controller on time, either via use of transverse coordinates [20] or through a zero dynamics manifold [28]. However, stabilization in the presence of contact uncertainty, particularly contact modes that are *not* part of the original plan, remains an open problem. While the QP controller used here reasons about the current contact state, the cost-to-go function from LQR provides no useful information in directions normal to the planned manifold. Numerical methods exist for formal stability analysis in the presence of impacts, such as in [25], though these tools do not yet scale to the dimensionality of these locomotion examples.

VI. Conclusion

In this work, we have presented a general purpose, end-to-end approach for synthesis and stabilization of optimal trajectories for robotic systems in contact with their environment. These contacts restrict motion of the robot to manifolds of feasible states, which can have complex geometries in multi-contact scenarios. By explicitly addressing the nature of these constraints, we design methods that seamlessly handle both non-minimal coordinates and underactuated dynamics, both of which present problems for many existing algorithms. The DIRCON algorithm is efficient, robust to initial seeds, and exhibits cubic integration accuracy. Use of lower order methods complicates the task of stabilization and can result in trajectories that do not accurately represent the true cost. As evidenced by the examples above, the combined LQR and QP control is capable of closely tracking these dynamic motions in terms of both state and control effort.

References

[1] A. D. Ames, K. Galloway, and J. W. Grizzle. Control Lyapunov Functions and Hybrid Zero Dynamics. In *Proceedings of the 51st IEEE Conference on Decision and Control*, Maui, HI, 2012.

[2] B. D. O. Anderson and J. B. Moore. *Optimal control: linear quadratic methods*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1990.

[3] J. T. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming*. SIAM Advances in Design and Control. Society for Industrial and Applied Mathematics, 2001.

[4] F. Bullo and A. D. Lewis. *Geometric Control of Mechanical Systems: Modeling, Analysis, and Design for Simple Mechanical Control Systems*. Texts in Applied Mathematics. Springer, Nov 4 2004.

[5] M. Buss, M. Hardt, J. Kiener, M. Sobotka, M. Stelzer, O. von Stryk, and D. Wollherr. Towards an Autonomous, Humanoid, and Dynamically Walking Robot. In *Proceedings of the 3rd International Conference on Humanoid Robotics*, pages 2491–2496, 2003.

[6] P. E. Crouch and R. Grossman. Numerical integration of ordinary differential equations on manifolds. *Journal of Nonlinear Science*, 3(1):1–33, 1993.

[7] H. Dai, A. Valenzuela, and R. Tedrake. Whole-body motion planning with centroidal dynamics and full kinematics. *IEEE-RAS International Conference on Humanoid Robots*, 2014.

[8] S. Feng, X. Xinjilefu, W. Huang, and C. G. Atkeson. 3D walking based on online optimization. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Atlanta, GA, Oct. 2013.

[9] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005.

[10] D. T. Greenwood. *Principles of dynamics*. Prentice-Hall Englewood Cliffs, NJ, 1988.

[11] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Science & Business Media, 2006.

[12] E. Hairer and G. Wanner. Solving ordinary differential equations ii: Stiff and differential-algebraic problems. *Springer series in computational mathematics*, 14, 1996.

[13] C. R. Hargraves and S. W. Paris. Direct trajectory optimization using nonlinear programming and collocation. *J Guidance*, 10(4):338–342, July-August 1987.

[14] A. Herzog, L. Righetti, F. Grimminger, P. Pastor, and S. Schaal. Momentum-based Balance Control for Torque-controlled Humanoids. *CoRR*, abs/1305.2042, 2013.

[15] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, Sept. 2003.

[16] T. Koolen, T. de Boer, J. Rebula, A. Goswami, and J. Pratt. Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models. *The International Journal of Robotics Research*, 31(9):1094–1113, Aug. 2012.

[17] T. Koolen, J. Smith, G. Thomas, S. Bertrand, J. Carff, N. Mertins, D. Stephen, P. Abeles, J. Englsberger, S. McCrory, J. van Egmond, M. Griffioen, M. Floyd, S. Kobus, N. Manor, S. Alsheikh, D. Duran, L. Bunch, E. Morphis, L. Colasanto, K.-L. H. Hoang, B. Layton, P. Neuhaus, M. Johnson, and J. Pratt. Summary of Team IHMC's Virtual Robotics Challenge Entry. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Atlanta, GA, Oct. 2013.

[18] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake. Optimization-based locomotion planning, estimation, and control design for the Atlas humanoid robot. *Autonomous Robots*, 40(3):429–455, 2016.

[19] S. Kuindersma, F. Permenter, and R. Tedrake. An efficiently solvable quadratic program for stabilizing dynamic locomotion. In *Proceedings of the International Conference on Robotics and Automation*, Hong Kong, China, May 2014. IEEE.

[20] I. R. Manchester. Transverse dynamics and regions of stability for nonlinear hybrid limit cycles. *Proceedings of the 18th IFAC World Congress, extended version available online: arXiv:1010.2241 [math.OC]*, Aug-Sep 2011.

[21] H. N. McClamroch and D. Wang. Feedback stabilization and tracking of constrained robots. *Automatic Control, IEEE Transactions on*, 33(5):419–426, 1988.

[22] K. D. Mombaur. Using optimization to create self-stable human-like running. *Robotica*, 27(3):321–330, 2009.

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

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

[25] M. Posa, M. Tobenkin, and R. Tedrake. Stability analysis and control of rigid-body systems with impacts and friction. *IEEE Transactions on Automatic Control (TAC)*, PP(99), 2015.

[26] C. D. Remy. *Optimal Exploitation of Natural Dynamics in Legged Locomotion*. PhD thesis, ETH ZURICH, 2011.

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

[28] Sreenath, K., Park, H.W., Poulakakis, I., Grizzle, and JW. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL. *International Journal of Robotics Research*, 2010.

[29] 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.

[30] R. Tedrake. Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems. `http://drake.mit.edu`, 2014.

[31] R. Tedrake, S. Kuindersma, R. Deits, and K. Miura. A closed-form solution for real-time ZMP gait generation and feedback stabilization. In *Proceedings of the International Conference on Humanoid Robotics*, 2015.

[32] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris. *Feedback Control of Dynamic Bipedal Robot Locomotion*. CRC Press, Boca Raton, FL, 2007.

[33] W. Xi and D. C. Remy. Optimal gaits and motions for legged robots. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3259–3265. IEEE, 2014.