Trajectory optimization for domains with contacts using inverse dynamics

Tom Erez and Emanuel Todorov[1]

Abstract— **This paper presents an algorithm for direct trajectory optimization in domains with contact. Since the contact introduces non-smooth dynamics, many standard algorithms of optimal control and reinforcement learning cannot be directly applied to such domains. We use a smooth contact model that can compute inverse dynamics through the contact, thereby avoiding hybrid representation of the non-smooth contact state. This allows us to formulate an unconstrained, continuous trajectory optimization problem, which can be solved using standard optimization tools. We demonstrate our approach by optimizing a running gait for a 31-dimensional simulated humanoid. The resulting gait is demonstrated in a movie attached as supplementary material. Most notably, the optimization result exhibits a synchronous motion of the arm and the opposite leg, eliminating undesired angular momentum; this is a key feature of bipedal running, and its emergence attests to the power of the optimization process.**

I. Introduction

Optimal control (or model-based reinforcement learning) allows us to derive motor behavior from first principles. The user encodes a high-level task in terms of a cost function, and provides a model of the dynamics; the optimization algorithm is responsible for discovering the low-level realization of the task, finding the control that yields a trajectory of minimal total cost.

Global methods of optimal control find an optimal policy over an entire volume of state space. Such methods are subject to the curse of dimensionality, as the volume of state space grows exponentially with the number of state dimensions. This motivates the study of trajectory optimization — by identifying only locally-optimal trajectories, these algorithms achieve polynomial scaling with the dimensionality of state space.

Trajectory optimization methods fall into two broad classes: *sequential* and *simultaneous* methods. Sequential methods optimize the control sequence by integrating the dynamics forward in time. These methods require a *forward* formulation of the dynamics, where the next state is computed from the current state and control. In contrast, simultaneous methods optimize the state trajectory itself, treating the dynamics as constraints that the trajectory must satisfy. While the forward dynamics can be used in direct methods, this formulation can also use *inverse* dynamics formulations (Section III-B), where the current control is computed from the current and next state.

*This work was supported by the Sackler Foundation

[1]T.E. and E.T are with the Department of Computer Science and Engineering, University if Washington, Seattle, USA {etom, todorov} at cs.washington.edu

However, most optimal control methods (either local or global, forward or inverse, simultaneous or sequential) can only solve domains with smooth dynamics. The most pertinent example of non-smooth dynamics are the collisional effects of unilateral constraints such as joint limits and contacts. This is a well-known difficulty for optimal control applications, especially in the realm of multi-joint robotics, where the optimal behavior often involves contact interaction with the environment.

The common approach to optimal control with contacts involves a hybrid modeling of the domain, introducing additional variables that explicitly represent the discrete contact state. However, most hybrid-based optimization methods require an a-priori specification of the discrete contact states; this constrains the search space, and requires the user to engage directly with the low-level realization of the task (Section II). In contrast, the method presented here allows us to formulate an unconstrained optimization where all contact states can be considered equally.

Since forward dynamics rely on numerical integration, small time-steps are necessary to prevent divergence when the dynamics are stiff or non-smooth (for example, Wang et al. [1] simulate walking with a time-step of $1/2400$s). This presents a significant challenge to algorithms of optimal control, which often scale at least linearly with the number of time-steps. In contrast, inverse dynamics does not rely on numerical integration, and is therefore immune to such problems of numerical instability, allowing for the representation of the same trajectory using a smaller number of time-steps. This makes inverse-dynamics an attractive design choice for optimal control methods.

However, inverse dynamics in domains with contact can easily become ill-defined. Consider, for example, a trajectory where a foot reaches a halt at zero height above the ground; the kinematics of this trajectory can have two rivalling dynamic interpretations: the agent voluntarily used strong forces to stop the foot's motion just above the ground, or the ground reaction forces caused the foot to stop. When contact is modeled discontinuously, these two dynamical interpretations are indistinguishable. However, these two cases will incur very different costs, since one involved much more effort than the other.

This can be resolved by using a smooth contact model; for example, when contact is modeled as a spring-damper system, the reaction forces are manifested by a compression of the spring, allowing a distinction between the two dynamical interpretations and enabling the computation of the inverse dynamics. However, in practice such models can pose

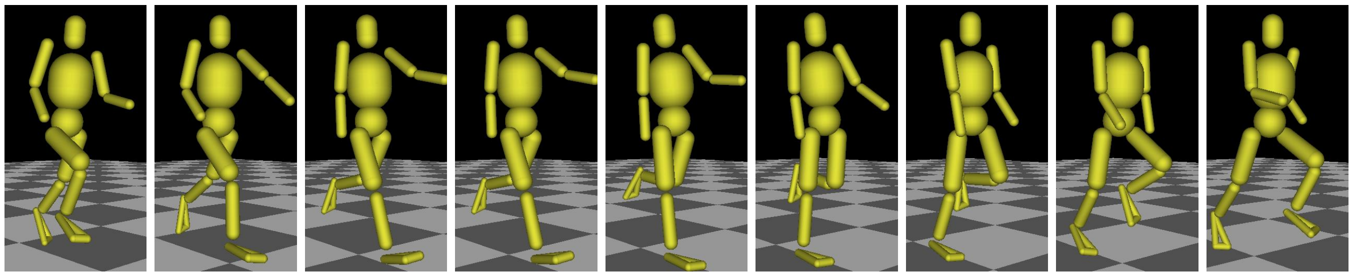

Fig. 1. A sequence of frames depicting the optimal running gait; this gait can be seen in a movie attached as supplementary material. Note the extension of the arm in frame 2, coinciding with the flight phase. This arm motion emerged in the optimal solution as a way to eliminate excessive angular momentum.

tremendous numerical difficulties for multi-contact systems, since the spring-damper parameters can be hard to tune. In this paper, we compute the reaction forces using a smooth, invertible contact model which is based on convex optimization (Section IV). This model allows us to formulate the contact-driven optimization problem in general terms, using only the continuous trajectory representation, and avoiding hybrid modeling or other contact-specific machinery.

We demonstrate the efficacy of this algorithm by solving a domain of bipedal running. This is an interesting domain because it includes a severely-underactuated flight phase, where the agent has no way to influence the trajectory of its center of mass. This makes it difficult to apply traditional gait design approaches (such as ZMP) to bipedal running. We optimize the motion pattern of a simulated 3D humanoid with 31 degrees of freedom, which is prohibitively-large for most standard optimization schemes. Our algorithm finds an optimal gait in about 10 minutes of computation time running on a standard desktop computer (Section VII). The resulting gait is best illustrated by a movie, included as supplementary material. Most notably, a coordinated movement of arms and legs emerges, as the upper body motion is used to balance the angular momentum created by leg swing.

II. Related work

Dynamics with contacts are a profound challenge to optimal control, and the standard way to overcome this obstacle is by using a *hybrid* representation. A discrete set of states is included in state-space, representing the boolean contact state of every possible contact pair. Given the discrete contact state, the continuous dynamics can be computed by treating the contact as a constraint. Additional computation is used to ensure that the contact forces remain inside the friction cone, adjusting the discrete contact state as necessary.

Hybrid modeling requires explicit representation of the contact state and significantly increases the burden of modeling the dynamics. For example, Bessonnet et al. [2] use inverse dynamics to optimize a gait pattern of a 3D robotic model. However, the dynamic model they use is not in general form, and their method requires an analytic formulation of a ZMP condition; therefore, their method cannot be easily extended to other morphologies, or a richer set of contacts. In another example, Ren et al. [3] study a planar model of human walking, yet their model cannot compute ground reaction forces during double support, and these must be approximated via data-driven heuristics.

In order to overcome the limitation of pre-specifying the contact state, Kim et al. [4] apply inverse dynamics to a 3D biomechanical hybrid model by explicitly representing and optimizing the timing of each gait phase (single-support vs. double-support). Mombaur et al. [5] also present an optimization method which solves simultaneously for body poses and contact times.

Others, such as Posa and Tedrake [6], incorporate the reaction forces themselves into the optimization problem. This side-steps the need to pre-specify the discrete contact state, but results in a complex optimization problem. Similarly, the contact smoothing technique used here allows the various gait phases to be optimized implicitly as part of the general motion pattern, and requires no special-purpose representation. However, our contact model allows us to represent only the trajectory itself, and resolve the contact forces as part of the inverse dynamics computation.

It is important to note that while we employ inverse dynamics, our goal is to generate an optimal trajectory from first principles. This is an orthogonal task to the wide literature on inverse dynamics control (e.g., [7]), where the goal is often to track a reference trajectory, using inverse dynamics to identify the adequate control signals.

III. Formulating the inverse dynamics

A. Minimal representation

In order to represent the mechanical state of a multibody system, we need to specify both the positions and velocities along some generalized coordinates. However, in simultaneous trajectory optimization, the velocities are represented implicitly, by virtue of representing consecutive positions. Therefore, it is redundant to represent the trajectory in all dimensions of state space (including both position and velocity). This redundancy leads to potential incongruence, which is manifested as an implicit set of constraints between the various variables of our search space. This gives rise to a Hessian which is dominated by those over-specified directions, which results in a slower optimization procedure (since self-inconsistency is penalized independently at every step).

In order to eliminate the mutual dependencies, we wish to perform the same simultaneous optimization using a *minimal* representation. Since every consecutive pair of positions implicitly defines a velocity, we can rewrite the same optimization problem while explicitly representing only the positions around the limit cycle; this will not only reduce the number of dimensions, but more importantly — eliminate mutual dependencies between the dimensions of the search space.

B. Computing the inverse dynamics

We assume that the state $\mathbf{x} \in \mathbb{R}^n$ is composed (at least) of a set of generalized coordinates $\mathbf{q} \in \mathbb{R}^q$ and their respective velocities $\dot{\mathbf{q}}$. In order to keep the syntax simple, we focus on the case where $\mathbf{x} = [\mathbf{q}^\mathsf{T}\ \dot{\mathbf{q}}^\mathsf{T}]^\mathsf{T}$ (so $n = 2q$); extending the optimization to additional state dimensions (such as muscle activation level or fiber length) can be done by defining the inverse dynamics of these quantities. The system is driven by a control signal $\mathbf{u} \in \mathbb{R}^m$; we are interested in the case of *underactuation* ($m < n$), where some state dimensions can only be affected indirectly.

Given a general-form cost function $\ell(\mathbf{x}, \mathbf{u})$, we seek to compute the locally-optimal N-step trajectory which minimizes the total cumulative cost $\sum_k \ell(\mathbf{x}^k, \mathbf{u}^k)$. This optimization takes place over the minimally-represented search space $\mathbf{Q} = \mathrm{stack}\{\mathbf{q}^k\} \in \mathbb{R}^{qN}$.

The dynamics express a relation between the current state $(\mathbf{q}^k, \dot{\mathbf{q}}^k)$ and the velocity at the next time-step $\dot{\mathbf{q}}'$:

$$\dot{\mathbf{q}}' = \dot{\mathbf{q}} + hM^{-1}(\mathbf{r} + \hat{\mathbf{u}}), \tag{1}$$

where h is the time-step, $\mathbf{q}$ is the configuration, $M = M(\mathbf{q})$ is the mass-matrix, $\mathbf{r} = \mathbf{r}(\mathbf{q}, \dot{\mathbf{q}})$ is the vector of total Coriolis, centripetal and other intrinsic forces, and $\hat{\mathbf{u}}$ is the *full* control (i.e., the control signal in the fully-actuated system). During forward simulation, $\hat{\mathbf{u}} = B\mathbf{u}$, where B is the $n \times m$ matrix determining which degrees of freedom are actuated.

The use of the full control $\hat{\mathbf{u}}$ is unavoidable during inverse dynamics — since we explicitly represent the entire trajectory, we must cope with trajectories that are infeasible in the underactuated system; however, the system controlled by $\hat{\mathbf{u}}$ is fully-actuated, and therefore every trajectory is dynamically-consistent by definition.

The full state-space trajectory can be easily recovered from a minimal representation. However, identifying the corresponding control sequence is not as simple. Forward dynamics answers the question "given the current position, velocity, and torque, what is the next position and velocity?", mapping from $(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u})$ to $\dot{\mathbf{q}}'$; inverse dynamics asks the question "given the current position and velocity, as well as the *next* velocity, what is the torque that is applied?", mapping from $(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}')$ to $\hat{\mathbf{u}}$.

It is convenient to define the *expanded* representation $\boldsymbol{\varphi}$ as a concatenation of these vectors: $\boldsymbol{\varphi} \triangleq [\mathbf{q}^\mathsf{T}\ \dot{\mathbf{q}}^\mathsf{T}\ \dot{\mathbf{q}}'^\mathsf{T}]^\mathsf{T}$. Note that the expanded representation $\boldsymbol{\varphi}$ is a linear function of three consecutive positions along a minimally-represented trajectory: $\boldsymbol{\varphi} = \Upsilon[\mathbf{q}^\mathsf{T}\ \mathbf{q}'^\mathsf{T}\ \mathbf{q}''^\mathsf{T}]^\mathsf{T}$, where `prime` ($'$) stands for "next time-step" as in equation 1, and:

$$\Upsilon = \begin{bmatrix} \mathbf{I} & 0 & 0 \\ -\mathbf{I}/h & \mathbf{I}/h & 0 \\ 0 & -\mathbf{I}/h & \mathbf{I}/h \end{bmatrix} \tag{2}$$

C. Helper forces

The inverse dynamics computes the full control $\hat{\mathbf{u}}(\boldsymbol{\varphi})$. In order to identify the control of the original underactuated system, we decompose $\hat{\mathbf{u}}$ into the sum of two terms, one representing the standard actuation and one representing the *helper* forces that apply to the originally-unactuated dimensions. Given the m-rank control matrix B of the original underactuated system, we can construct a second matrix $\breve{B}$ whose columns span the kernel of B. This allows us to uniquely decompose $\hat{\mathbf{u}}$:

$$\hat{\mathbf{u}}(\boldsymbol{\varphi}) = B\mathbf{u}(\boldsymbol{\varphi}) + \breve{B}\breve{\mathbf{u}}(\boldsymbol{\varphi}). \tag{3}$$

We may now use the original (underactuated) control signal $\mathbf{u}$ in the computation of $\ell(\mathbf{x}, \mathbf{u})$ as before. In order to prevent reliance on helper forces, we use a penalty method and compute a cost term on $\breve{\mathbf{u}}$ which constrains it to 0; in the experiments below we use a quadratic cost $\frac{1}{2}\gamma||\breve{\mathbf{u}}||^2$, resulting in the performance criterion:

$$\beta(\boldsymbol{\varphi}) = \ell\big(\mathbf{x}(\boldsymbol{\varphi}), \mathbf{u}(\boldsymbol{\varphi})\big) + \frac{1}{2}\gamma||\breve{\mathbf{u}}(\boldsymbol{\varphi})||^2. \tag{4}$$

where $\mathbf{x}(\boldsymbol{\varphi})$ is shorthand for truncating the next velocity from $\boldsymbol{\varphi}$, leaving only the current state $\mathbf{x} = [\mathbf{q}^\mathsf{T}\ \dot{\mathbf{q}}^\mathsf{T}]^\mathsf{T}$ (section III-B).

D. Invariant coordinates

In order to optimize a limit cycle, $\mathbf{Q}$ must represent a closed trajectory. However, the case of locomotion presents a challenge — when all generalized coordinates are considered, locomotion is not a limit cycle but rather a coil-shaped trajectory, because some positional state increases with every period. This complication can be avoided by excluding this coordinate from the state space, keeping only the corresponding velocity. This is allowed because the dynamics of the system is *invariant* in that dimension, and so we may arbitrarily set the value of this coordinate to 0 during dynamics computations. When the velocities are implicitly represented by the positions $\mathbf{q}$ (as is the case here), we modify $\mathbf{Q}$ to explicitly represent the velocity of this coordinate. The size of $\mathbf{Q}$ remains the same, but the transformation matrix Υ is modified to copy the velocity explicitly from $\mathbf{Q}$ to $\boldsymbol{\varphi}$, and insert 0 at the relevant entry of $\boldsymbol{\varphi}$.

E. Symmetry

In some periodic domains (such as legged locomotion) we expect the optimal limit cycle to be symmetric — for example, the starting position of the swing leg should match the ending position of the stance leg, and vice versa. This constraint allows us to optimize for only one step, instead of a full stride. We enforce this constraint by applying a linear transformation (swapping the sides of the body) to the last state $\mathbf{q}^N$ whenever it is considered in conjunction with the first state $\mathbf{q}^1$.

IV. Inverse dynamics with contacts

Contact modeling is a complex task, since the reaction forces must satisfy friction cone constraints. Furthermore, it is physically-unrealistic to allow reaction forces between distant objects. One established approach is to model the contact through a Linear Complementarity Problem (LCP), where action-at-a-distance is eliminated by making distance and reaction force mutually-exclusive: if the distance in contact coordinates is greater than zero (i.e., no contact), the contact force must be zero.

However, LCP cannot be used to compute inverse dynamics, because the trajectory does not contain enough information to recover both the reaction forces and the actuation. For example, consider two consecutive states where a leg rests on the ground; any actuation that comes short of lifting the leg might not have any kinematic effect, and yet the cost should increase due to the effort exerted. Furthermore, mutual exclusivity leads to non-smooth dynamics, which pose a significant difficulty to trajectory optimization. The theory of Stochastic LCP [8] leads to a relaxation of the complementarity, as it demonstrates that in the presence of stochasticity, the *expected* reaction forces (and hence the dynamics) are smooth; unfortunately, the SLCP contact model is also non-invertible.

This paper makes use of a different algorithm for computing smooth reaction forces [9]. This contact model uses a relaxed form of the complementarity condition (see below), resulting in smooth dynamics; most importantly, the model can differentiate between the effects of the agent's action and the contact reaction forces, allowing us to compute the inverse dynamics even in the presence of contacts.

Given the current position $\mathbf{q}$, a geometric computation can yield J, the Jacobian between state coordinates and contact coordinates. We use this Jacobian to introduce the reaction force $\mathbf{p}$ (specified in contact coordinates) to equation 1:

$$\dot{\mathbf{q}}' = \dot{\mathbf{q}} + hM^{-1}(\mathbf{r} + \hat{\mathbf{u}}) + M^{-1}J^{\mathsf{T}}\mathbf{p}.$$

Note that the last term is *impulsive*, in that it is not multiplied by the time step h. Using J again to project this equation to contact space, we get:

$$J\dot{\mathbf{q}}' = \mathbf{c} + A\mathbf{p},$$

where $\mathbf{c} = J\big(\dot{\mathbf{q}} + hM^{-1}(\mathbf{r} + B\mathbf{u})\big)$ is the velocity in contact space when $\mathbf{p} = 0$, and $A = JM^{-1}J^{\mathsf{T}}$ is the inverted mass matrix in contact coordinates. A *forward* contact model maps from $(\mathbf{c}, A)$ to $\mathbf{p}$ (and therefore $\dot{\mathbf{q}}'$); an *inverse* contact model maps from $(\dot{\mathbf{q}}', A)$ to $\mathbf{p}$ (and therefore $\hat{\mathbf{u}}$).

Our contact model [9] solves for $\mathbf{p}$ by seeking the reaction force that minimizes kinetic energy in contact space. Following the derivation presented there, this condition is equivalent to minimizing some convex residual (see below). The friction cone constraints are translated to log-barrier functions $d(\mathbf{p})$, and the problem is solved with a modified interior-point method.

The residual has the form $r(\mathbf{p}) = \frac{1}{2}\mathbf{p}^{\mathsf{T}}(A + R)\mathbf{p} + \mathbf{p}^{\mathsf{T}}\mathbf{c}$, where $R = R(J\mathbf{q})$ is a regularization matrix which serves for ensuring (relaxed) complementarity: R is a diagonal matrix whose elements are a function of the current distance in contact coordinates. When the distance along a certain contact dimension is large, the corresponding terms in R grow rapidly, and the corresponding terms in the minimizing $\mathbf{p}$ reduce to 0; when the contact distance is small (or negative, indicating penetration), the corresponding terms in R go to zero, and the contact model is free to choose $\mathbf{p}$ that eliminates all kinetic energy in that dimension.

As our prior work shows [9], this convex optimization can be inverted: since $\mathbf{p}$ minimizes $r + d$, it must satisfy $r_{\mathbf{p}} + d_{\mathbf{p}} = 0$. Through some further analytic manipulation, we can construct a residual expression for $\mathbf{p}$ that depends only on A and $J\dot{\mathbf{q}}'$. Unsurprisingly, the resulting residual includes the term $\frac{1}{2}\mathbf{p}^{\mathsf{T}}R\mathbf{p}$, allowing R to manifest a relaxed complementarity condition as before. The parametric form of R is not dictated by the theory; in our experiments, we got the most realistic-looking results with R being exponentially-dependent on the contact distance.

V. Inverse dynamics optimization

We seek to minimize the total cost $\boldsymbol{\beta} = \sum_k \beta^k(\boldsymbol{\varphi}^k)$ by computing $\boldsymbol{\beta}_{\mathbf{Q}}$ and $\boldsymbol{\beta}_{\mathbf{QQ}}$, its derivatives WRT our minimal representation of the trajectory.

Using the expanded representation $\boldsymbol{\varphi}$ (section III-B), we define the stack of expanded representation vectors $\boldsymbol{\Phi} \triangleq \mathrm{stack}\{\boldsymbol{\varphi}^k\} \in \mathbb{R}^{3qN}$, which is a linear function of the minimal representation $\mathbf{Q}$:

$$\boldsymbol{\Phi} = \Lambda\mathbf{Q}, \tag{5}$$

where the matrix $\Lambda \in \mathbb{R}^{qN \times 3qN}$ is made of adjacent copies of Υ (equation 2).

We can compute the gradient and Hessian of β WRT $\boldsymbol{\varphi}$ by finite-differencing the inverse-dynamics function $\mathbf{u}(\boldsymbol{\varphi})$ and applying the chain rule. This provides a quadratic model for β for every time step k along the trajectory:

$$\beta(\boldsymbol{\varphi}^k + \delta\boldsymbol{\varphi}) \approx \beta_0^k + \delta\boldsymbol{\varphi}^{\mathsf{T}}\beta_{\boldsymbol{\varphi}}^k + \frac{1}{2}\delta\boldsymbol{\varphi}^{\mathsf{T}}\beta_{\boldsymbol{\varphi}\boldsymbol{\varphi}}^k\delta\boldsymbol{\varphi}.$$

We stack the gradients $\boldsymbol{\beta}_{\boldsymbol{\Phi}} = \mathrm{stack}\{\beta_{\boldsymbol{\varphi}}^k\}$ and arrange the quadratic terms in the block-diagonal matrix $\boldsymbol{\beta}_{\boldsymbol{\Phi\Phi}} = \mathrm{diag}\{\beta_{\boldsymbol{\varphi}\boldsymbol{\varphi}}^k\}$ to obtain a quadratic model for $\boldsymbol{\beta}$ over the expanded representation:

$$\boldsymbol{\beta}(\boldsymbol{\Phi} + \delta\boldsymbol{\Phi}) \approx \boldsymbol{\beta}_0 + \delta\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\beta}_{\boldsymbol{\Phi}} + \frac{1}{2}\delta\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\beta}_{\boldsymbol{\Phi\Phi}}\delta\boldsymbol{\Phi}$$

We can now compress the derivatives of $\boldsymbol{\beta}$ WRT the expanded representation to their equivalents in the minimal representation using the matrix Λ (equation 5):

$$\boldsymbol{\beta}_{\mathbf{Q}} = \Lambda^{\mathsf{T}}\boldsymbol{\beta}_{\boldsymbol{\varphi}} \tag{6a}$$

$$\boldsymbol{\beta}_{\mathbf{QQ}} = \Lambda^{\mathsf{T}}\boldsymbol{\beta}_{\boldsymbol{\varphi}\boldsymbol{\varphi}}\Lambda. \tag{6b}$$

VI. Compressed representation

Equations 6 use the sparse matrix Λ to linearly transform a quadratic model in the expanded representation $\mathbf{\Phi}$ to the corresponding quadratic model in the minimal representation $\mathbf{Q}$. This reduction in dimensionality is information-preserving, because the expanded representation is only a notational convenience. However, this manipulation suggests that a similar manipulation can be used to allow optimization with compressed representation that is smaller than $\mathbf{Q}$, as long as it is linearly-expandable to $\mathbf{Q}$. This criterion is met by many low-dimensional representations of closed trajectories, such as Fourier transform and splines, and this type of dimensionality reduction is common in simultaneous optimization [2], [10].

In the experiments described below we use a Fourier series representation for compression. In one dimension, we can linearly expand a vector $\mathbf{t} \in \mathbb{R}^s$ (representing the first s coefficients in a Fourier series) to the N-vector $F\mathbf{t}$ representing the full periodic trajectory (the Fourier-transform matrix $F \in \mathbb{R}^{N\times s}$ can be obtained, for example, by taking the first s rows of an FFT of the identity matrix). Given $\mathbf{S} \in \mathbb{R}^{sq}$, the Fourier series of all state dimensions, we write: $\mathbf{Q} = \mathcal{F}\mathbf{S}$, where $\mathcal{F} \in \mathbb{R}^{sq\times Nq}$ is the full-trajectory Fourier-transform matrix, built from stacked copies of the matrix F. This martix allows us to write the derivatives of $\boldsymbol{\beta}$ WRT the Fourier representation:

$$\boldsymbol{\beta}_{\mathbf{S}} = \mathcal{F}^{\mathsf{T}}\boldsymbol{\beta}_{\mathbf{Q}} \tag{7a}$$

$$\boldsymbol{\beta}_{\mathbf{SS}} = \mathcal{F}^{\mathsf{T}}\boldsymbol{\beta}_{\mathbf{QQ}}\mathcal{F}. \tag{7b}$$

Although this is a lower-dimensional representation, the computational cost of every iteration remains almost the same, because the derivatives of the inverse dynamics still require finite-differencing of the expanded representation $\mathbf{\Phi}$. Note that this method can only find the locally-optimal trajectory within the limited subspace of all trajectories; the optimal unconstrained trajectory $\mathbf{Q}^*$ still has a lower cost than $\mathbf{S}^*$.

VII. Results

We demonstrate the power of inverse optimization by optimizing a walking gait for a simulated 3D bipedal robot, illustrated in Figure 1. The model has 31 degrees of freedom: the ball joint between the torso and the pelvis has 3 DOF (the head is welded to the torso), each leg has 3 DOF at the hip, 1 at the knee and 3 at the ankle, and each arm has 3 DOF at the shoulder and 1 at the elbow. The physical properties of the body parts are specified in table I.

Every foot has three potential contact points with the ground: one at the back, and two at the front. In this experiment we use $N = 21$ timesteps of 15msec, which correspond to ~ 190 steps per minute, in accordance with standard human running pace. The Fourier representation uses $s = 11$, in accordance with Ren et al.[3].

The cost function of the running task compares the horizontal velocity of the runner's center of mass $\dot{\mathbf{q}}_y$ to some desired velocity, while maintaining all body parts at a forward orientation:

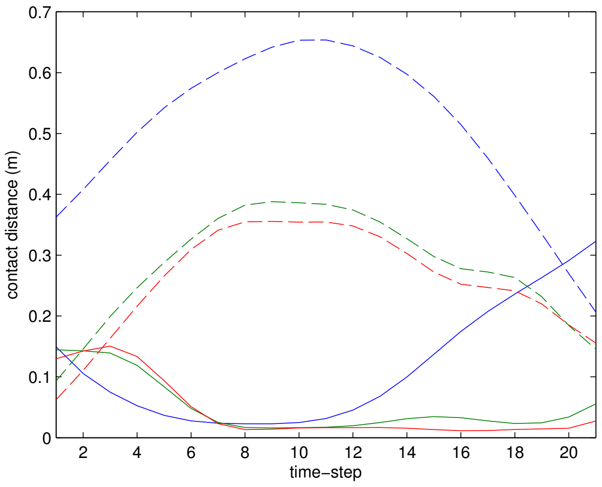

Fig. 2. Distance in contact space between feet and ground. Time-step is 15msec, resulting in a period of 0.315s, or about 3 steps per second.

$$\ell(\mathbf{x},\mathbf{u}) = 100(\dot{\mathbf{q}}_y - v_d)^2 + ||\mathbf{q}_{xz}|| + 0.01||\mathbf{u}||^2$$

where $\mathbf{q}_{xz}$ is a vector of all joint angles in the $x-$ and $z-$axes, and the last term is a quadratic cost on applied joint torques. The penalty term on $\mathbf{q}_{xz}$ is required because we are using direct torque control, and in the absence of such penalty the runner tries to distribute the actuation across all spatial angles equally, resulting in a hideously-unrealistic gait (for example, the controller attempts to balance the hip rotation in the lateral and forward directions, creating great lateral excursions of the thigh). In our simulation we use a desired velocity v_d = 6m/s, and the resulting optimal running gait reaches an average velocity of 5.1m/s.

The optimization process converges after about 2500 Newton-step iterations, which take about 10 minutes on a desktop computer with two 6-core processors. The most computationally-intensive step is finite-differencing the inverse dynamics. However, this computation can be executed in parallel, allowing us to harness parallel architectures and cluster processing.

The resulting gait can be seen in a movie which is included as supporting material. Figure 2 shows the distances, in contact space, of the feet's contact points during one step. The heel strikes at time-step 6, and the toes soon follow at step 8, which makes this a heel-first running gait. With the heel leaving the ground at step 12, and toe-off at time 21, the flight phase extends between time-steps 1 and 5, making it a veritable running gait.

The most striking feature of the resulting running gait is the emergent coordination between the legs and the opposite arms. The left arm thrusts forward in synchrony with the right leg during the flight phase (best seen in frame 2 of Figure 1), eliminating the angular momentum of the pelvis and torso that would otherwise result in an increased cost. This feature came about without any form of explicit mod-

body	length (cm)	radius (cm)	mass (kg)
trunk	10	18	28.5
head	10	8.5	4.2
pelvis	13	n/a	9.2
thigh	35	7.4	7.3
shin	31	5	2.8
foot	27.4	3	2.3
upper arm	30	5	2.7
lower arm	26	4	1.5

TABLE I
MORPHOLOGICAL SPECIFICATION OF THE 3D RUNNER. ALL BODIES BUT PELVIS ARE CAPSULES. TOTAL HEIGHT: 1.9M ($\sim$6'3"); TOTAL WEIGHT: 75.25KG ($\sim$166 LBS.).

eling, and attests to the power of the optimization process. The arm also jerks in the anti-phase of the forward thrust; we are not sure what causes that motion.

VIII. DISCUSSION

This paper uses an invertible smooth contact model to find an optimal trajectory in a high-dimensional domain. The formulation we use is completely general, and requires no hybrid representation. While we demonstrate this algorithm in a domain of humanoid bipedal running, the algorithm we present has general applicability, and the very same optimization can be applied to a diverse set of tasks, from multi-legged locomotion to dexterous hand manipulation. In particular, while domains of legged locomotion design often rely on some simplified model such as SLIP [11], our optimization made no use of domain-specific heuristics.

The resulting optimal motion shares some features of human bipedal running. However, biomechanical realism is not the primary goal of this project. We are certain that further tweaking of the cost function and the body anthropometry can lead to running gaits that more faithfully imitate human running. Yet, such biomimicry is contingent on the development of algorithms capable of optimizing behavior in biomechanically-realistic domains (which are often high-dimensional) in the presence of contacts. The algorithm presented in this paper meets these requirements.

We are not explicitly representing the stochasticity of the domain. However, following the theory of Stochastic LCP (section IV), the smooth contact dynamics can be construed as an approximation for the expected reaction forces under stochastic dynamics. In order to control a real-world system (or a stochastic simulation), the Hessian around the optimal trajectory can be used to derive a locally-linear feedback controller [12]. By invoking the *separation* principle [13], the feedback controller optimized for the deterministic domain can serve to control its stochastic counterpart, as long as the noise profile is not dependent on the state.

We take a model-based approach to optimizing behavior, which allows for efficient optimization. In contrast, model-free methods construct an approximation of the dynamics (either in forward or inverse formulation) from interaction with the environment. This approach cannot be directly applied to tasks which require powerful articulated motion, and result in inherently unstable behavior. However, it is impossible to build a perfectly accurate model of a multi-joint robot. Therefore, in order to apply such model-based methods to control real robots, some mechanism for model adaptation and learning must be incorporated. One example for such a method was presented by Abbeel et al. [14]. In addition, some tolerance for modeling errors must be incorporated into the feedback controller; this can be achieved, for example, using model-predictive control [15].

REFERENCES

[1] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Optimizing walking controllers," in *ACM Transactions on Graphics (TOG)*, ser. SIGGRAPH Asia '09. New York, NY, USA: ACM, 2009.

[2] G. Bessonnet, J. Marot, P. Seguin, and P. Sardain, "Parametric-Based dynamic synthesis of 3D-Gait," *Robotica*, vol. 28, no. 04, pp. 563–581, 2010.

[3] L. Ren, R. K. Jones, and D. Howard, "Predictive modelling of human walking over a complete gait cycle," *Journal of Biomechanics*, vol. 40, no. 7, pp. 1567–1574, 2007.

[4] H. J. Kim, Q. Wang, S. Rahmatalla, C. C. Swan, J. S. Arora, K. Abdel-Malek, and J. G. Assouline, "Dynamic motion planning of 3D human locomotion using gradient-based optimization," *Journal of Biomechanical Engineering*, vol. 130, no. 3, June 2008.

[5] K. Mombaur, A. Truong, and J.-P. Laumond, "From human to humanoid locomotionan inverse optimal control approach," *Autonomous Robots*, vol. 28, pp. 369–383, 2010.

[6] M. Posa and R. Tedrake, "Direct trajectory optimization of rigid body dynamical systems through contact," 2012 (Under review).

[7] L. Sentis and O. Khatib, "A whole-body control framework for humanoids operating in human environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 2641–2648.

[8] Y. Tassa and E. Todorov, "Stochastic complementarity for local control of discontinuous dynamics," in *Proceedings of Robotics: Science and Systems (RSS)*, 2010.

[9] Emanuel Todorov, "A convex, smooth and invertible contact model for trajectory optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[10] F. C. Anderson and M. G. Pandy, "Dynamic optimization of human walking," *Journal of Biomechanical Engineering*, vol. 123, no. 5, pp. 381–390, Oct. 2001.

[11] M. H. Raibert, *Legged Robots that Balance*. MIT Press, 1986.

[12] Y. Tassa, T. Erez, and E. Todorov, "Optimal limit-cycle control recast as bayesian inference," in *Proceedings of thh IFAC world congress*, 2011.

[13] R. F. Stengel, *Optimal Control and Estimation*. Dover Publications, Sept. 1994.

[14] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, "An application of reinforcement learning to aerobatic helicopter flight," in *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*, 2007, p. 1.

[15] T. Erez, Y. Tassa, and E. Todorov, "Infinite-horizon model predictive control for periodic tasks with contacts," in *Proceedings of Robotics: Science and Systems (RSS)*, 2011.