

Robust Physics-Based Locomotion Using Low-Dimensional Planning

Igor Mordatch Martin de Lasa Aaron Hertzmann

University of Toronto

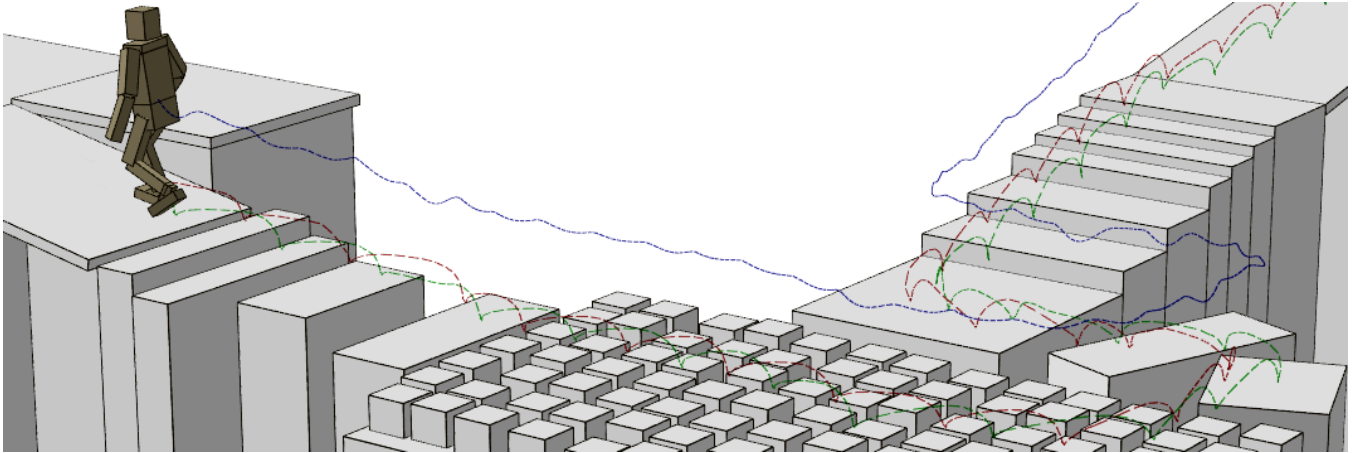


Figure 1: Interactive locomotion control over varied terrain. Gait, footsteps, and transitions are automatically generated, based on user-specified goals, such as direction, step length, and step duration. In the above example, a user steers the biped across uneven terrain with gaps, steps, and inclines.

Abstract

This paper presents a physics-based locomotion controller based on online planning. At each time-step, a planner optimizes locomotion over multiple phases of gait. Stance dynamics are modeled using a simplified Spring-Load Inverted (SLIP) model, while flight dynamics are modeled using projectile motion equations. Full-body control at each instant is optimized to match the instantaneous plan values, while also maintaining balance. Different types of gaits, including walking, running, and jumping, emerge automatically, as do transitions between different gaits. The controllers can traverse challenging terrain and withstand large external disturbances, while following high-level user commands at interactive rates.

CR Categories: I.3.1 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

Keywords: Physics-Based Animation, Control, Locomotion

1 Introduction

Generating robust locomotion control of physically-simulated characters is very challenging. At every instant, control must select forces that affect not only immediate state, but also future motion

<http://www.dgp.toronto.edu/~mdelasa/slip>

completion. This is especially important when moving across irregular terrain with little margin for error. In these situations, foot step locations and modes of travel must be jointly selected to set up the next step. For example, a character walking on flat ground may need to switch to a different gait, such as running or jumping, when traversing stepping stones or chasms. Planning these types of motions remains an open problem. Performing a full spacetime optimization of the motion would be extremely expensive, and would not be able to handle unforeseen disturbances, changes to the environment, or new user-specified goals. As an alternative, simplified dynamics models that enable online planning can be used. However, existing models are gait specific and do not plan control over multiple foot-steps.

This paper introduces a locomotion controller for full-body 3D physics-based characters that plans joint torques by optimizing a low-dimensional physical model. At each instant, the controller optimizes a plan over multiple phases of locomotion. To generate the plan, full-body dynamics are approximated by a set of closed-form equations-of-motion (EOM), allowing replanning to occur at every time step. The controller then chooses optimal full-body torques according to the current plan, while maintaining balance. Depending on the task goals and terrain, different gaits, including walking, running, and jumping, and gait transitions, emerge naturally. When navigating over terrain with a limited number of footholds, our method automatically determines optimal paths. In the accompanying video, we show several examples of navigation over uneven terrain, including gaps, inclines, stairs, large drops, and stepping stones. We also demonstrate robustness to external disturbances and projectile avoidance capabilities. Our method requires no pre-processing and no motion capture data; a small set of user-level parameters allows control over the final motion.

2 Related Work

A great deal of work in physics-based animation has used joint-level PD control with finite-state machines to obtain a variety of walking, running, and athletic motions [Faloutsos et al. 2001; Hod-

gins et al. 1995; Laszlo et al. 1996; Raibert and Hodgins 1991; Yin et al. 2007]. These approaches require extensive iteration, new parameter settings for each type of motion, and have limited capabilities on uneven terrain.

Much recent work has focused on tracking motion capture reference data [Abe et al. 2007; da Silva et al. 2008a; da Silva et al. 2008b; Macchietto et al. 2009; Muico et al. 2009; Sok et al. 2007; Tsai et al. 2010; Yamane and Hodgins 2009]. Using example motions makes it much easier to specify motion style and can produce very high-quality results. However, such methods are only appropriate near example motions (or transformations of them), and thus require trajectories for every desired type and style of motion.

In biomechanics and robotics, simplified low-dimensional mechanical models have long been used to understand the essential dynamic properties of human motion [Alexander 1980; Collins et al. 2005; McGeer 1990; Srinivasan and Ruina 2006]. Hence, such models are attractive as a basis for designing controllers. However, doing so presents a number of challenges. The Inverted Pendulum Model (IPM), which models center-of-mass (COM) motion during single stance, has been used to control walking [Kajita et al. 2001; Kajita et al. 2003; Pratt et al. 2006; Rebula et al. 2007; Tsai et al. 2010]. The IPM is limited by a constant leg-length, is constrained to have exactly one foot on the ground at all times, and does not model swing leg behavior. Hence, it cannot model double-stance or ballistic motion. The Spring-Loaded Inverted Pendulum (SLIP) [Full and Koditschek 1999; Schwind and Koditschek 2000] generalizes the IPM by replacing the fixed length leg with a spring, thereby capturing energy storage and release during running. Since exact SLIP EOM are non-integrable, they are costly to evaluate. To date, analysis of SLIP and IPM models has been limited to understanding steady-state cyclic behavior [Schwind and Koditschek 2000] or for devising simple control rules and metrics [Pratt and Tedrake 2005; Pratt et al. 2006; Rebula et al. 2007]. We propose a closed-form SLIP-based model that captures motion across multiple phases, selects footholds producing good subsequent steps, and generates a continuum of behaviors including walking, running, and jumping.

Using low-dimensional models for control requires a method to map the low-dimensional control outputs to all the joints in the full character. To do this, most methods formulate a weighted Quadratic Program (QP), that trades-off between mocap and control target tracking [Abe et al. 2007; da Silva et al. 2008a; da Silva et al. 2008b; Macchietto et al. 2009]. Tsai et al. [2010] propose a velocity-based LCP formulation. Ramamoorthy and Kuipers [2008] use offline manifold learning to perform this mapping. All these methods require motion capture data to disambiguate redundancy. Our method does not rely in any way on motion capture data. Our system employs a full-body control optimization designed to maintain balance while following the low-dimensional plan. The low-dimensional plan is used only to guide the full-body controller, not to define it. Because plans are inexpensive to evaluate, any deviations between the motion of the full character and the simplified planning model are corrected in subsequent planner solutions.

Many previous methods use planning to help bipeds traverse uneven or constrained terrain. Though kinematic planning suffices for highly stable actions (e.g., [Kuffner et al. 2003]), more dynamic motions require kinodynamic planning. Chestnutt [2007] describes planners for robotic locomotion across terrains with dynamic obstacles and Coros et al. [2009] demonstrate robust controllers that can perform a variety of level-ground tasks. These methods assume the existence of predefined walking controllers capable of performing footstep actions; a high-level planner or policy then chooses sequences of controllers by a discrete search. Our approach is significantly different. We define a single controller that determines accelerations at every instant by performing trajectory optimization.

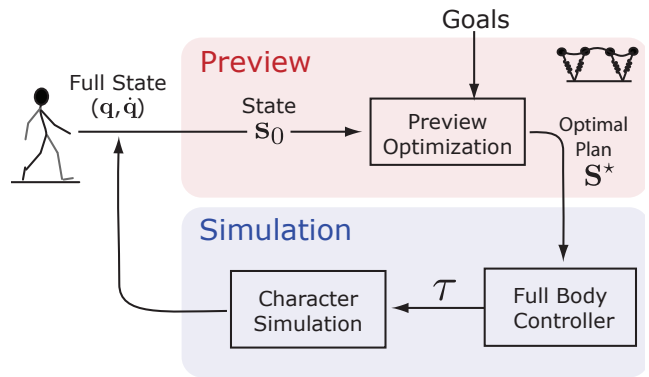


Figure 2: System Overview. At each time step, character state $\mathbf{q}, \dot{\mathbf{q}}$ is mapped to a lower-dimensional preview model state, \mathbf{s}_0 . Next, a preview optimization generates an optimal plan \mathbf{S}^* that meets user-specified goals. Lastly, the full-body controller optimization calculates joint torques $\boldsymbol{\tau}$, based on the instantaneous accelerations from \mathbf{S}^* . Torques are applied to the simulation, resulting in a new character state.

Many different types of gaits and terrain-navigation abilities arise automatically from this controller, without needing to be created separately in advance. We do not attempt long-term planning.

3 Overview

We describe a locomotion controller that calculates joint torques $\boldsymbol{\tau}$ for a physics-based character (Fig. 2). At each instant of the simulation, a plan \mathbf{S}^* is computed by nonlinear optimization of the motion over the subsequent two footsteps. Since trajectory optimization [Witkin and Kass 1988] for full-body locomotion remains very computationally expensive [Liu et al. 2005; Wampler and Popović 2009], we optimize the trajectory of a low-dimensional preview model instead. Full-body torques are then chosen such that the full-character’s COM follows instantaneous plan accelerations.

The preview model is based on the Spring-Loaded Inverted Pendulum (SLIP) [Full and Koditschek 1999] (Sec. 4). A fixed preview schedule divides the next two footsteps into 6 phases: double stance, left single-stance, flight, double stance, right single-stance, and flight. The optimization includes objectives to achieve a target goal (e.g., desired step length and heading) and objectives to compensate for simplifications (e.g., foot position relative to hip). The optimization produces different gaits by omitting certain phases: omitting double-stance leads to jogging, while omitting flight leads to walking. The plan is computed by optimizing control parameters \mathbf{U} , for the SLIP motion \mathbf{S} during these 6 phases. We make simplifying approximations to derive closed-form simulation of the SLIP dynamics. The optimization yields an optimal plan \mathbf{S}^* .

Using the optimal plan \mathbf{S}^* , joint torques $\boldsymbol{\tau}$ are determined by a second optimization (Sec. 6). This optimization attempts to have the full character match the instantaneous accelerations of $\ddot{\mathbf{S}}^*$, while also maintaining balance. Given torques $\boldsymbol{\tau}$, a simulator computes joint accelerations $\ddot{\mathbf{q}}$ and contact forces \mathbf{f}_c , and updates the full-character state $(\mathbf{q}, \dot{\mathbf{q}})$ by integration. This process of replanning and torque optimization is repeated at every time-step. Instead of replanning every simulation timestep, somewhat faster performance can also be obtained by replanning less frequently.

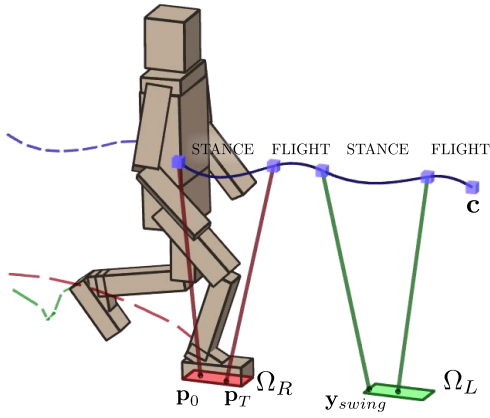


Figure 3: Low-dimensional planning. At each instant, a low-dimensional plan \mathbf{S}^* is computed, specifying COM (\mathbf{c}), and COP, ($\mathbf{p}_0 / \mathbf{p}_T$) motion, as well as the next foot plant (\mathbf{y}_{swing}). Motion is divided into stance and flight phases. A running motion schedule is shown above, with footplants $\Omega_{R/L}$.

4 Preview Simulation

This section describes the low-dimensional model used to simulate dynamics over multiple phases of motion. Stance phases are represented with a modified SLIP model, with simplifications that allow closed-form integration. The resulting motion over multiple phases is piecewise polynomial, allowing analytic integration of objective function terms (Sec. 5). Ballistic phases are represented by COM motion only. Initial conditions for the preview simulation come from the current full-body character state, which provides initial COM position \mathbf{c}_0 and velocity $\dot{\mathbf{c}}_0$, initial heading α_0 and angular velocity $\dot{\alpha}_0$, and right/left foot contact polygons $\Omega_{R/L}$. α_0 is set to the orientation of the pelvis about the vertical axis.

4.1 Stance Model

When the character has one or more feet on the ground, a Spring Loaded Inverted Pendulum (SLIP) model [Full and Koditschek 1999] is used to describe character motion. We modify the SLIP in several ways. First, we add an explicit representation of the support polygon. Second, since exact SLIP equations must be numerically integrated, we use an approximate model that decouples COM motion in directions parallel and perpendicular to the ground plane. This allows our equations of motion and objective functions to be computed in closed-form. Lastly, rather than assuming a fixed COP position, we allow it to move within the support polygon.

The SLIP model is parameterized by a COM position \mathbf{c} and a heading α , representing orientation of the pelvis about the vertical axis. At any given instant, the complete DOFs of the model are:

$$\mathbf{s}(t) = [\mathbf{c}^T(t) \quad \dot{\mathbf{c}}^T(t) \quad \alpha(t) \quad \dot{\alpha}(t) \quad \Omega_R \quad \Omega_L]^T \quad (1)$$

where $\Omega_{R/L}$ are the support regions for the right/left foot respectively. The support regions are constant throughout a given phase. When a foot is in the air, its support region is empty ($\Omega = \emptyset$). Otherwise, its support region is modeled as a quadrilateral.

The complete set of single and double stance control parameters used by our model, are summarized by an 8-dimensional vector:

$$\mathbf{u}_s = [T \quad \mathbf{p}_0^T \quad \mathbf{p}_T^T \quad r_0 \quad r_T \quad \ddot{\alpha}]^T. \quad (2)$$

The duration of the phase is set by the parameter T . The SLIP model is controlled by a spring that connects the COM, \mathbf{c} , to COP,

\mathbf{p} , on the ground. The spring has rest length $r(t)$ and a fixed stiffness k . Rather than fixing the COP as is typically done (e.g., [Kajita et al. 2001]), we allow a time-varying COP. Walking with a fixed COP corresponds to walking on pin-like feet and does not take advantage of the foot's whole support region. In humans, foot rolling is an important factor in the production of smooth, energy-efficient walking [Adamczyk et al. 2006]. Both the rest length and COP are controlled linearly:

$$\mathbf{p}(t) = \frac{t}{T}(\mathbf{p}_T - \mathbf{p}_0) + \mathbf{p}_0 \quad (3)$$

$$r(t) = \frac{t}{T}(r_T - r_0) + r_0 \quad (4)$$

where $\mathbf{p}_{0/T}$ are the start/end COP locations, and $r_{0/T}$ are the start/end rest lengths. Angular motion is controlled by setting a constant angular acceleration $\ddot{\alpha}$.

4.2 Stance Dynamics and Simulation

We now describe SLIP EOM used to obtain $\mathbf{s}(t)$ (Eq. 1), given initial conditions \mathbf{s}_0 and control \mathbf{u} . Horizontal, vertical, and rotational motion are decoupled to obtain closed-form solutions. Components are combined:

$$\mathbf{c}(t) = [x(t) \quad y(t) \quad z(t)]^T. \quad (5)$$

All derivations assume flat ground with axes x/y and z being parallel and perpendicular to the ground plane respectively. Equations are represented in world coordinates and converted to polynomial form using fifth-order Taylor approximation.

Our approximate SLIP model results from decoupling and linearizing the inverted pendulum model of Kajita et al. [2001]. In the vertical direction, we also add a spring with constant stiffness. This produces a model with a straightforward interpretation: horizontal motion is governed by inverted pendulum dynamics, while vertical motion is described by spring-mass dynamics.

Horizontal Motion. We first derive EOM along the horizontal x -axis. EOM in the y direction are identical. Assuming an inverted pendulum of mass m , a height h above the ground, and an angle θ from the vertical axis, gravitational forces act in the horizontal direction as $mg \tan \theta$. Using $\tan \theta = (x - p_x)/h$, we have:

$$m\ddot{x} = mg(x - p_x)/h. \quad (6)$$

Because COP motion is linear (Eq. 3), the solution is given by:

$$x(t) = \beta_1 e^{\alpha t} + \beta_2 e^{-\alpha t} + p_x(t) \quad (7)$$

where $\beta_1 = (x_0 - p_{0,x})/2 + (\dot{x}_0 T - (p_{T,x} - p_{0,x}))/2\alpha T$, $\beta_2 = (x_0 - p_{0,x})/2 - (\dot{x}_0 T - (p_{T,x} - p_{0,x}))/2\alpha T$. $\alpha = \sqrt{g/h}$, x_0/\dot{x}_0 are the horizontal COM position/velocity at the beginning of the stance phase, and h is the COM height at the beginning of the stance phase.

Vertical motion. Vertical motion is modeled using a spring mass system, subject to gravitational and spring forces:

$$m\ddot{z} = k(r - z) - mg. \quad (8)$$

Because the rest length varies linearly (Eq. 4), the motion is given by:

$$z(t) = d_1 \cos(\omega t) + d_2 \sin(\omega t) + r(t) - g/\omega^2 \quad (9)$$

where $d_1 = z_0 - r_0 + g/\omega^2$, $d_2 = \dot{z}_0/\omega - (r_T - r_0)/(T\omega)$, $\omega = \sqrt{k/m}$, and z_0/\dot{z}_0 are the COM position/velocity at the beginning of the stance motion in the vertical direction. In all our examples we use $g = 9.81 \text{ m/s}^2$ and $k = 1$.

	1	2	3	4	5	6
	DS	SS	F	DS	SS	F
Walking						
Running						
Standing						

Figure 4: Preview optimization schedule. Depending on the optimization results, different phases are excluded: Walking alternates between double-stance (DS) and single-stance (SS). Running alternates between SS and Flight (F). Standing only uses a single DS phase. Exchange of support (EOS) occurs between subsequent stance phases.

Heading. To compute the character’s heading $\alpha(t)$, we assume the pelvis orientation is independent of linear COM motion. Because the controlled angular acceleration $\ddot{\alpha}$ is constant, we have:

$$\alpha(t) = \alpha_0 + \dot{\alpha}_0 t + \frac{1}{2} \ddot{\alpha} t^2. \quad (10)$$

4.3 Flight Dynamics and Simulation

The complete state vector during flight is:

$$\mathbf{s}(t) = [\mathbf{c}(t)^T \quad \dot{\mathbf{c}}(t)^T \quad \alpha(t) \quad \dot{\alpha}(t) \quad \emptyset \quad \emptyset]^T \quad (11)$$

When the character is in flight, projectile motion equations are used to describe COM motion. Since joint forces have no effect on the COM trajectory or heading, the only control parameter for the flight phase is the duration T :

$$\mathbf{u}_f = [T] \quad (12)$$

The equations of motion are then:

$$\mathbf{c}(t) = \mathbf{c}_0 + \dot{\mathbf{c}}_0 t - \frac{1}{2} \ddot{\mathbf{c}} t^2 \quad (13)$$

$$\alpha(t) = \alpha_0 + \dot{\alpha}_0 t \quad (14)$$

and $\ddot{\mathbf{c}} = [0 \quad g \quad 0]^T$ is the acceleration due to gravity.

4.4 Multi-step simulation

Simulation over multiple steps requires concatenating multiple stance and flight phases. Walking emerges as a sequence of alternating double and single-stance phases, whereas running alternates between single-stance and flight (Fig. 4). In all cases, single-stance phases alternate between left-foot and right-foot stance. We capture multiple types of locomotion with a single, fixed preview schedule of 6 phases (Fig. 4): double-stance, single-stance, flight, double-stance, single-stance, and flight. Optimizing a preview schedule entails optimizing six sets of control parameters:

$$\mathbf{U} = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \mathbf{u}_3 \quad \mathbf{u}_4 \quad \mathbf{u}_5 \quad \mathbf{u}_6]. \quad (15)$$

In every schedule, only a subset of the phases will occur. Phases assigned a zero duration by the optimization ($T_i = 0$), will be omitted from the final plan. The initial phase must match the initial state of the fully-body character, e.g., if the current character state is single-stance, then the first double-stance phase is skipped. Because it is not possible to transition from flight directly to double-stance (i.e., one foot will always land slightly before the other), at least one of phases 3 and 4 will always be skipped.

When the character is not standing in place, exchange-of-support between left and right stance feet will occur after phases 2 or 3. This requires a new footplant to be selected. We use an additional 2D control parameter:

$$\mathbf{u}_{foot} = \mathbf{y}_{swing} \quad (16)$$

to specify where the swing foot should be planted. Given \mathbf{y}_{swing} and the character’s heading α at the beginning of the corresponding phase, the shape of the character’s foot is projected onto the environment to determine the corresponding Ω for rest of the schedule.

Given initial conditions, \mathbf{s}_0 and control parameters, \mathbf{U} , a preview simulation is then performed by applying EOM for each phase sequentially. Concatenating the individual trajectories gives the low-dimensional motion over the entire preview schedule:

$$\mathbf{S}(t) = [\mathbf{s}_1 \quad \mathbf{s}_2 \quad \mathbf{s}_3 \quad \mathbf{s}_4 \quad \mathbf{s}_5 \quad \mathbf{s}_6] \quad (17)$$

Since each phase is polynomial, the complete sequence is piecewise polynomial. The total duration of the schedule is the sum of the individual durations, $T_{sched} = \sum_i T_i$.

To avoid impulsive changes in motion, we enforce continuity in control parameters across consecutive stance phases. Specifically, \mathbf{p}_T for phase i must equal \mathbf{p}_0 for phase $i + 1$; likewise r_T for phase i must equal r_0 for phase $i + 1$. As a result, the preview control parameters \mathbf{U} consists of 23 free parameters.

5 Preview Optimization

Given initial conditions \mathbf{s}_0 , the preview optimization selects optimal control parameters, \mathbf{U}^* for the preview schedule, relative to a set of user-specified energy terms $g_i(\mathbf{S})$:

$$\mathbf{U}^* = \arg \min_{\mathbf{U}} \sum_i w_i g_i(\mathbf{S}) \quad (18)$$

Because the landscape of our objective has many local minima, we solve the optimization using Covariance Matrix Adaptation (CMA) [Hansen 2006]. To speed up the optimization, we reuse the solution from the previous timestep as the initial guess for the current solution. In our implementation, we run CMA with a maximum of 5000 objective function evaluations and a population size of 15. The projectile avoidance examples, discussed in Section 7, uses up to 10000 evaluations. Two types of terms are included in the preview optimization objective: goal objectives and modeling objectives. The optimal control parameters \mathbf{U}^* yield an optimal simulation \mathbf{S}^* .

Our system uses the simulator described in [de Lasa et al. 2010], which uses an inelastic ($\epsilon_{rest} = 0$) LCP-based contact model with Coulomb friction ($\mu = 1$). Our human model has 37 degrees of freedom (Fig. 3). All limbs are assumed to have constant density. Table 1 lists typical parameters used in our system to achieve different types of motion.

5.1 Goal Objectives

Using goal objectives, users can specify desired properties of motion. Our system uses four types of goal objectives: i) step duration, ii) step length, iii) pelvis heading, and iv) COM height. Throughout this section, we denote the first and second foot step of the schedule using A and B respectively. See Table 1 for typical settings used in our examples.

We seek solutions with steps of user-specified duration T_{step} . Deviations are penalized using:

$$g_{steptime} = (T_A - T_{step})^2 + (T_B - T_{step})^2 \quad (19)$$

where the total duration of each step, T_A and T_B , are the sums of their respective double and single-stance durations.

To encourage steps of a particular length, d_{step} , and in the desired direction of travel, \mathbf{d} , we use:

$$g_{stepdist} = (d_A - d_{step})^2 + (d_B - d_{step})^2 \quad (20)$$

where $d_A = (\mathbf{c}_A - \mathbf{c}_0)^T \mathbf{d}$ and $d_B = (\mathbf{c}_B - \mathbf{c}_A)^T \mathbf{d}$ are the length of the first and second step, projected in the desired direction of travel, and \mathbf{c}_i is the Cartesian location of the COM at different points in the preview.

Since heading changes cannot be achieved instantaneously, we determine error in orientation at the end of the preview using:

$$g_{heading} = (\alpha_B - \alpha_d)^2 \quad (21)$$

where α_d is the desired character heading. This term allows a user to steer the character.

Transitions between walking and jumping arise based on foothold selection and user constraints. To encourage motions with pronounced ballistic phases, we include an objective:

$$g_{com} = (h_\epsilon - h_d)^2 \quad (22)$$

$$h_\epsilon = \max(h_{apex} - h_{start}, h_{apex} - h_{end}). \quad (23)$$

h_{start} , h_{end} , and h_{apex} are the COM heights at the start, end, and apex of the first flight phase, respectively. h_d is expressed relative to the COM standing height.

5.2 Modeling Objectives

Modeling objectives play an important role in compensating for simplifications made by the decoupled preview dynamics model. We use three modeling objectives: i) COM acceleration, ii) leg length, and iii) foot position relative to hips. Although modeling objectives use error integrals over the preview window, these quantities can be calculated in closed-form, due to the piecewise-polynomial COM preview motion (Sec. 4.2).

Because we optimize a simplified passive model, with no explicit force inputs, it does not make sense to measure stride power. Instead, we seek smooth motions with small COM accelerations:

$$g_{accel} = \int \ddot{\mathbf{c}}(t) dt. \quad (24)$$

A key simplification made by our stance model is that COM height is constant during stance. Though this simplifies the mathematical preview model description, it can produce unnatural ‘‘flat walking’’ motions. Natural walking is characterized by an arc-like motion of the stance leg about the COP. To encourage this type of motion, we penalize motions with large variations from the target leg length:

$$g_{leg} = \sum_{i \in \{R, L\}} \int (L_i^2 - L_r^2)^2 dt \quad (25)$$

where $L_i = \|\mathbf{y}_{hip}^i - \mathbf{y}_{ankle}^i\|$ is the length of leg i , measured between the hip and the ankle (Fig. 3) and L_r is the character’s leg length when standing. The positions \mathbf{y}_{hip} and \mathbf{y}_{ankle} are defined by fixed horizontal offsets from \mathbf{c} and \mathbf{y}_{swing} , respectively. This term also helps avoid drift, keeping the leg length ‘‘centered’’ near its nominal value. Weights for this objective term are kept small, since motions such as running and jumping require substantial leg compression.

Table 1: Controller Parameter Values. Nominal values are based on observed human motion. Typical preview optimization weights are: $w_{steptime} = 10$, $w_{stepdist} = 10$, $w_{heading} = 1$, $w_{com} = 0$, $w_{accel} = 0.05$, $w_{leg} = 10$, $w_{hip} = 10$.

Action	d_{step} [m]	T_{step} [s]	w_{com}	h_d [m]
Walk	0.75	0.6	0	0
Run/Jump	1.05	0.4	> 0	0 – 0.5

In our preview model, we assume pelvis orientation in the axial plane is completely decoupled from forward motion. Even for characters with small hips, changes to pelvis orientation will require leg-length changes. To compensate for this simplification and to generate motions with the feet roughly pointed in the same direction as the pelvis, we use an objective:

$$g_{hip} = \sum_{i \in \{R, L\}} \int \|\mathbf{y}'_{hip} - \mathbf{y}'_{ankle}\|^2 dt \quad (26)$$

where \mathbf{y}'_{hip} and \mathbf{y}'_{ankle} are the 2D projections of the hip and ankle position onto the ground plane. This term encourages the hip to stay over the ankle.

5.3 Constraints

To ensure solutions calculated for the simplified model are valid when mapped to the full character, several geometric constraints, based on target character skeletal properties, are enforced. With the exception of leg length, all of the constraints are expressed in CMA as variable bounds constraints. Leg length constraints are represented using a sigmoidal soft-constraint [Liu et al. 2005].

The COP locations $\mathbf{p}_{0/T}$ must lie within the base-of-support during each stance phase. The locations $\mathbf{p}_{0/T}$ are represented by bilinear coordinates within the contact quadrilateral. During double-stance, a quadrilateral is fit to $\Omega_L \cup \Omega_R$.

We limit the model’s telescoping leg length:

$$L_i \in [L_{min}, L_{max}] \quad (27)$$

to ensure leg length limits imposed by the full character are not violated. L_{max} is the maximal hip to ankle distance of the full character when the legs are fully extended and L_{min} is the same distance when the legs are fully contracted.

The footstep location is constrained to lie near the COM:

$$\|\mathbf{c}_{\parallel} - \mathbf{y}_{swing}\|_1 < \frac{1}{2} L_r \quad (28)$$

where \mathbf{c}_{\parallel} are the COM coordinates projected on the ground plane and L_r is the character’s leg length when standing. The L_1 norm is used to enable this to be expressed as bounds constraints.

Each phase must have a non-negative duration $T \geq 0$.

6 Full-Body Controller

Once \mathbf{S}^* has been selected, a second optimization computes full-body joint torques $\boldsymbol{\tau}$ for the current time instant. The only quantities used from \mathbf{S}^* are the instantaneous second derivatives ($\ddot{\mathbf{c}}_d, \ddot{\alpha}$) at the start time $t = 0$, and the target footplant \mathbf{y}_{swing} .

Given the optimal preview motion \mathbf{S}^* , we calculate:

$$\mathbf{x} = [\boldsymbol{\tau}^T \quad \ddot{\mathbf{q}}^T \quad \boldsymbol{\lambda}^T]^T \quad (29)$$

where τ are the full-body control torques, $\ddot{\mathbf{q}}$ is the generalized accelerations of the body, and λ are linearized friction-cone basis weights, used to estimate contact forces. We formulate control synthesis as a quadratic program (QP) subject to dynamics constraints:

$$\begin{aligned} \mathbf{x}^* = \arg \min_{\mathbf{x}} \sum_i w_i E_i(\mathbf{x}) \\ \text{subject to } C(\mathbf{x}) = 0, \mathbf{D}\mathbf{x} + \mathbf{f} \geq 0 \end{aligned} \quad (30)$$

where $E_i(\mathbf{x})$ are quadratic energy terms and constraints enforcing physical realism. At each time instant, the optimal \mathbf{x} is recalculated; the resulting accelerations $\ddot{\mathbf{q}}$ are then integrated. The dynamics constraints, character model, and integration are implemented as in [de Lasa et al. 2010]. We do not use prioritized optimization [de Lasa et al. 2010], although we suspect that it could be used to provide better stylistic control and robustness.

The optimization uses several objectives. The first two objectives aim to have the COM and pelvis move in the direction suggested by the plan. To do this, we use objectives $E_{com} = \|\ddot{\mathbf{c}}_r - \ddot{\mathbf{c}}\|^2$ and $E_{heading} = (\ddot{\alpha}_r - \ddot{\alpha})^2$, where $\ddot{\mathbf{c}}_r$ and $\ddot{\alpha}_r$ are computed by analytic differentiation of the current polynomial plan representation. Feet are moved to the target footholds, \mathbf{y}^{swing} , using our target objective E_{swing} [de Lasa et al. 2010]. Unlike in that work, it is not necessary to hand-design gait-specific COM trajectories or foot targets.

Another objective, E_{hip} , implemented as a setpoint objective [de Lasa et al. 2010], favours motions with similar hip-to-COM distance as the simplified model. Nominal hip-to-COM distance is calculated at the beginning of the simulation, when the character is in its reference standing posture, and assumed to remain constant. This helps us come up with a better measure of the leg length commanded by the SLIP. This quantity is expressed in the heading reference frame, to encourage the legs and feet to point in the desired direction of travel.

Additional objective and stabilization terms are taken from [de Lasa et al. 2010]: angular momentum regulation (E_{AM}), rest pose (E_{rest}), head stabilization (E_{head}), and foot contact during stance ($E_{contact}$). We do not include stylistic terms, such as arm torque minimization, as we found that these can cause problems during sharp turns. Angular momentum (AM) regulation performs two roles in this optimization. First, as in [de Lasa et al. 2010], AM regulation is useful for balance and stability. Additionally, because our simplified model does not represent rotational motion outside the axial plane, it cannot accurately preview motion with significant AM. Hence, keeping AM small for the full-body movement helps prevent deviations between the models.

It is also possible to compute the optimal preview less frequently, since, in the absence of unexpected events, the preview optimization is normally consistent across a large number of frames. This gives some improvement in performance. In addition to planning at integer multiples of the integration timestep, we find it useful to allow replanning at contact events (e.g., change of support).

7 Results

Using the described formulation, we generate a diverse set of robust locomotion behaviors. Generating distinct behaviors, such as walking, running, and standing, as well as transitions between these behaviors, is achieved by specifying only a high-level set of task goals. No state machines, special treatment of transitions, or motion capture data are required to generate these motions. Resulting motions are shown in the accompanying video.

Performance. When updating dynamics and control at 250Hz, our unoptimized implementation runs at approximately 15% real-

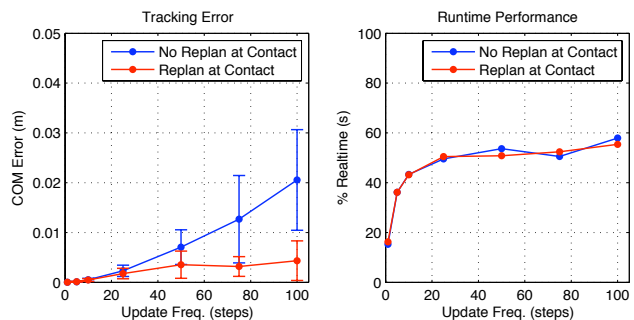


Figure 6: Performance vs. Update Frequency. Mean COM tracking error and runtime speed, as a function of replanning frequency, for level-ground walking. Standard deviations are shown for tracking error. COM error is the mean cumulative COM tracking error over each replanning window. Planning was performed at specified multiples of the integration timestep (blue). To improve stability, we also allowed replanning at contact transitions (red). Replanning only at contacts yields speedups with only minimal tracking error.

time on a Intel i7 920 2.67 GHz processor. Planning less frequently significantly improves performance. To quantify impact of planning frequency on level-ground walking, we test two algorithm variants (cf. Fig. 6). First, we restrict preview computation to occur every 1, 5, 10, 25, 50, 75, and 100 integration steps. At each step, a COM setpoint objective [de Lasa et al. 2010] is used with reference acceleration $\ddot{\mathbf{c}}_r$ computed by differentiating the plan, and with gain $k_p^{com} = 1000$. All other aspects of the control remain unchanged. Second, we repeat the first experiment, but also allow replanning at all contact events. Less-frequent planning significantly improves performance, but also decreased robustness. Without allowing replanning at contact events, we were not able to plan less frequently than every 100 steps. When limiting replanning to only occur at contact events, the average speed is 55% realtime; remaining computation is for our simulator and the full-body controller. Replanning only at contacts yields significant speedups with minimal motion changes. In this case, replanning constitutes a negligible portion of computation time. To further improve performance, our method might also benefit from more sophisticated optimization methods, such as Basin-CMA [Wampler and Popović 2009], which could leverage the analytically-differentiable objective provided by our method. Planning frequency could also be dynamically adjusted based on detected disturbances or preview deviations.

Standing. To have the character stand in place, we set the desired step-size $d_{step} = 0$ and deactivate the step-duration objective ($w_{steptime} = 0$). This results in a preview consisting of a single arbitrarily long double-stance phase. If the desired heading, α_d , is changed during standing, the character will take the necessary steps to avoid errors from g_{hip} , while also satisfying $g_{heading}$. Small changes in desired orientation are handled by the torso, whilst large turns coordinate leg joints. In addition to smoothing locomotion gaits, using a linear COP model (Eq. (4)) enhances standing by allowing perturbations to be handled using in-place weight shifts, rather than requiring many small compensatory steps.

Walking, Running, and Jumping. Walking and running are performed by adjusting step parameters (Table 1). Transitions between these two locomotion modes occurs automatically as parameters are changed. To force motion with marked ballistic periods, the COM height objective (Eq. (23)) is activated and h_d is set to a nonzero value. Alternatively, the weighting of g_{accel} can be decreased. This will allow larger deviations in $\ddot{\mathbf{c}}$.

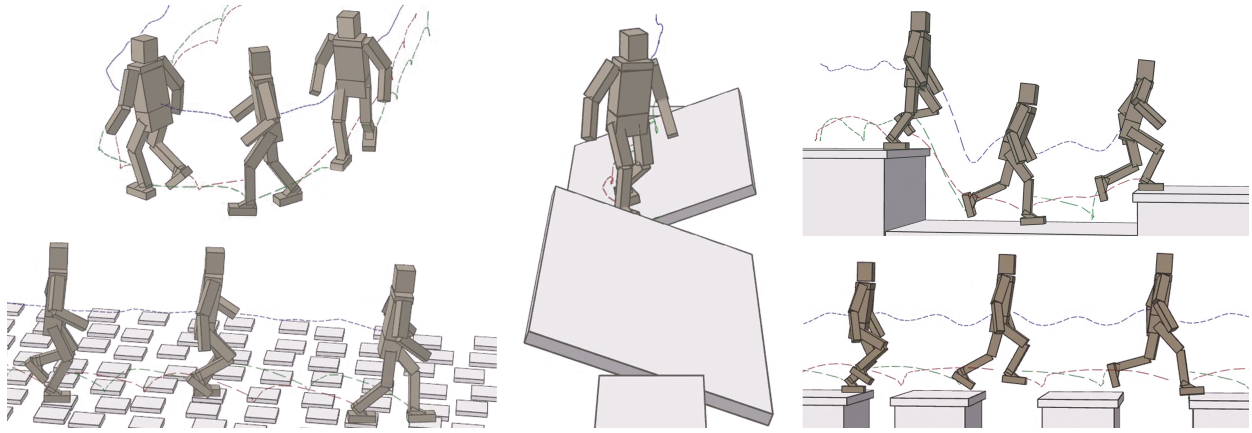


Figure 5: Uneven Terrain Capabilities. In addition to walking, running, and turning, low-dimensional planning allows traversal of a variety of terrains including large drops (1 m), steps (0.5 m), stepping stones, gaps (0.25-0.9 m), and inclines ($\pm 15^\circ$).

Uneven Terrain Navigation. Because our preview model plans into the future, control developed for flat ground can also be used for uneven terrain without changes. The controller can successfully traverse many types of uneven and constrained terrain, including gaps, inclines, stairs, large drops, and stepping stones (Figure 5). A demonstration of the character traversing an obstacle course with many of these elements is shown in Figure 1. Terrain is represented as a height field, which is provided to the preview optimization to select good footplants \mathbf{y}_{swing} . For large gaps, we have found it useful to provide the optimization with a signed distance field describing favored foot plant locations on the terrain. Specifically, when possible we prefer the character step in the middle of a platform, away from edges. Currently our system handles inclines of $\pm 15^\circ$. Larger inclines violate modeling assumptions of our approximate SLIP model, which assumes flat ground.

External Disturbances and Projectile Avoidance. Previous work has quantified controller robustness through external perturbation tests, e.g., throwing projectiles at the character [Abe et al. 2007; Coros et al. 2009]. In the accompanying video we demonstrate controller robustness, by testing external forces ranging from 100-500 N applied for 0.1-0.2 sec. to the character’s head, at random/cyclic intervals. In addition to applying forces when the character is in stance, we also test the challenging case of in-air disturbances. In both cases, the planner generates steps to avoid failure.

Our controller can also be robust to projectiles by dodging them. We add a projectile avoidance constraint to the planner that constrains the plan to keep the COM above a minimum distance D from the path of the projectile:

$$\min_t \|\mathbf{p}(t) - \mathbf{c}(t)\|^2 > D^2 \quad (31)$$

where t indexes the duration of the plan, and $\mathbf{p}(t)$ is the future path of the projectile. This constraint is enforced by a sigmoidal soft constraint, and the minimum is computed over a set of discrete sample times t . As a result, plans are generated that take necessary evasive manoeuvres to step away from projectiles. When the heading objective is deactivated, the character can change whole-body orientation to more quickly avoid projectiles. When heading is active, avoidance strategies generally involve taking larger steps.

8 Discussion

Attempting to directly apply nonlinear optimization methods to plan full-body dynamics motion is currently intractable. We show

that effective online motion planning of highly-dynamic behavior can be performed using a low-dimensional model. Careful choice of objectives allows us to optimize essential features of the high-dimensional character’s gait. These choices generate a wide range of motions over varied terrain. The controller is memoryless: no persistent state (e.g., index into a state machine or reference trajectory) is stored by the controller, with the exception of remembering the last stance foot during flight.

Our controllers cannot consistently handle certain types of motion. For example, we are unable to reliably generate 180° turns when the character is running quickly. This arises since the preview model assumes the body is a point mass and neglects rotational effects. The fixed, two foot-step horizon we currently use is also too short for certain terrain types. For example, the character has difficulty finding good footholds across gaps larger than 1 m and making anticipatory movements to walk on narrow beams. Toe-stubbing can occur when climbing up stairs larger than 0.5 m, since the current swing foot objective does not account for terrain collisions. Since we do not monitor inter-limb collisions, limb interpenetration can occur.

The focus of this work has been on robustness in challenging tasks and there is room for improvement in the style of motion. There are two main factors limiting the style. First, the SLIP model does not have knees, arms, and other elements that may affect style. Second, all of our examples err on the side of being overly stiff at all times. Although steady walking can be accomplished with fairly loose control, and strategies in [de Lasa et al. 2010] were designed with that in mind, the types of aggressive motions shown in this work require stiffer response. For example, if the arms are too loose, they can swing wildly during sharp turns. Varying full-body controller stiffness based on dynamic state or preview model predictions, is an interesting direction for future work. Using prioritized optimization may also decrease sensitivity of stylistic terms, as described in [de Lasa et al. 2010].

Our method opens up a number of interesting possibilities and directions for future research. By considering more sophisticated preview models, it should be possible to generate a broader range of motions. We currently use a fixed schedule for all generated motions, with fixed left/right-foot alternation. Hopping on one leg could be achieved by enforcing left-foot-only contacts; more general contact sequences could be optimized, as in [Wampler and Popović 2009]. Employing a low-dimensional model with angular momentum regulation would allow motion requiring inertia shaping, such as flips and cartwheels. Additional interactions between

characters and the environment or each other could be handled with additional representations of end-effectors.

Acknowledgments

The authors are indebted to Simon Breslav for his assistance with video production and to Jack Wang for inspiring technical discussions. We thank Nikolaus Hansen for his publicly available CMA implementation. This research is supported in part by NSERC, CFI, and the Ontario MRI. This work was done while AH was on a sabbatical visit at Pixar Animation Studios.

References

- ABE, Y., DA SILVA, M., AND POPOVIĆ, J. 2007. Multiobjective Control with Frictional Contacts. In *Proc. SCA*, 249–258.
- ADAMCZYK, P. G., COLLINS, S. H., AND KUO, A. D. 2006. The advantages of a rolling foot in human walking. *J. Exp. Biology* 209, 3953–3963.
- ALEXANDER, R. M. 1980. Optimum walking techniques for quadrupeds and bipeds. *J. Zoology* 192, 97–117.
- CHESTNUTT, J. 2007. *Navigation Planning for Legged Robots*. PhD thesis, Carnegie Mellon University.
- COLLINS, S., RUINA, A., TEDRAKE, R., AND WISSE, M. 2005. Efficient Bipedal Robots Based on Passive Dynamic Walkers. *Science* 307, 1082–1085.
- COROS, S., BEAUDOIN, P., AND VAN DE PANNE, M. 2009. Robust Task-based Control Policies for Physics-based Characters. *ACM Trans. Graphics* 28, 5, 170.
- DA SILVA, M., ABE, Y., AND POPOVIĆ, J. 2008. Interactive Simulation of Stylized Human Locomotion. *ACM Trans. Graphics* 27, 3, 82.
- DA SILVA, M., YEUHI, A., AND POPOVIĆ, J. 2008. Simulation of Human Motion Data using Short-Horizon Model-Predictive Control. *Proc. Eurographics* 27, 2.
- DE LASA, M., MORDATCH, I., AND HERTZMANN, A. 2010. Feature-Based Locomotion Controllers. *ACM Trans. Graphics* 29, 3.
- FALOUTSOS, P., VAN DE PANNE, M., AND TERZOPOULOS, D. 2001. Composable controllers for physics-based character animation. In *Proc. SIGGRAPH*, 251–260.
- FULL, R. J., AND KODITSCHKEK, D. E. 1999. Templates and anchors: Neuromechanical hypotheses of legged locomotion on land. *J. Exp. Biology* 202, 3325–3332.
- HANSEN, N. 2006. The CMA Evolution Strategy: A Comparing Review. In *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*. 75–102.
- HODGINS, J. K., WOOTEN, W. L., BROGAN, D. C., AND O'BRIEN, J. F. 1995. Animating human athletics. In *Proc. SIGGRAPH*, 71–78.
- KAJITA, S., MATSUMOTO, O., AND SAIGO, M. 2001. Real-time 3D walking pattern generation for a biped robot with telescopic legs. In *Proc. ICRA*, 2299–2306.
- KAJITA, S., KANEHIRO, F., KANEKO, K., FUJIWARA, K., HARADA, K., YOKOI, K., AND HIRUKAWA, H. 2003. Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point. In *Proc. ICRA*, 1620–1626.
- KUFFNER, J., NISHIKAWA, K., KAGAMI, S., INABA, M., AND INOUE, H. 2003. Motion Planning for Humanoid Robots. In *Proc. ISRR*, 365–374.
- LASZLO, J., VAN DE PANNE, M., AND FIUME, E. 1996. Limit cycle control and its application to the animation of balancing and walking. In *Proc. SIGGRAPH*, 155–162.
- LIU, C. K., HERTZMANN, A., AND POPOVIĆ, Z. 2005. Learning Physics-Based Motion Style with Nonlinear Inverse Optimization. *ACM Trans. Graphics* 24, 3, 1071 – 1081.
- MACCHIETTO, A., ZORDAN, V., AND SHELTON, C. 2009. Momentum Control for Balance. *ACM Trans. Graphics* 28, 3, 80.
- MCGEER, T. 1990. Passive Dynamic Walking. *Int. J. Robotics Research* 9, 2, 62–82.
- MUICO, U., LEE, Y., POPOVIĆ, J., AND POPOVIĆ, Z. 2009. Contact-aware Nonlinear Control of Dynamic Characters. *ACM Trans. Graphics* 28, 3, 81.
- PRATT, J., AND TEDRAKE, R. 2005. Velocity-Based Stability Margins for Fast Bipedal Walking. In *Fast Motions in Robotics and Biomechanics—Optimization and Feedback Control*, vol. 340, 299–324.
- PRATT, J., CARFF, J., DRAKUNOV, S., AND GOSWAMI, A. 2006. Capture Point: A Step toward Humanoid Push Recovery. In *Proc. Humanoid Robots*, 200–207.
- RAIBERT, M. H., AND HODGINS, J. K. 1991. Animation of dynamic legged locomotion. In *Proc. SIGGRAPH 1991*, 349–358.
- RAMAMOORTHY, S., AND KUIPERS, B. 2008. Trajectory Generation for Dynamic Bipedal Walking through Qualitative Model Based Manifold Learning. In *Proc. ICRA*, 359–366.
- REBULA, J., CANAS, F., PRATT, J., AND GOSWAMI, A. 2007. Learning Capture Points for Humanoid Push Recovery. In *Proc. Humanoid Robots*, 65–72.
- SCHWIND, W. J., AND KODITSCHKEK, D. R. 2000. Approximating the stance map of a 2-DOF monoped runner. *J. Nonlinear Science*, 533–568.
- SOK, K. W., KIM, M., AND LEE, J. 2007. Simulating Biped Behaviors from Human Motion Data. *ACM Trans. Graphics* 26, 3, 107.
- SRINIVASAN, M., AND RUINA, A. 2006. Computer optimization of a minimal biped model discovers walking and running. *Nature* 439, 7072, 72–75.
- TSAI, Y.-Y., LIN, W.-C., CHENG, K. B., LEE, J., AND LEE, T.-Y. 2010. Real-Time Physics-Based 3D Biped Character Animation Using an Inverted Pendulum Model. *Trans. Visualization and Computer Graphics* 16, 325–337.
- WAMPLER, K., AND POPOVIĆ, Z. 2009. Optimal Gait and Form for Animal Locomotion. *ACM Trans. Graphics* 28, 3, 60.
- WITKIN, A., AND KASS, M. 1988. Spacetime Constraints. In *Proc. SIGGRAPH*, vol. 22, 159–168.
- YAMANE, K., AND HODGINS, J. 2009. Simultaneous Tracking and Balancing of Humanoid Robots for Imitating Human Motion Capture Data. *Proc. IROS*, 2510–2517.
- YIN, K., LOKEN, K., AND VAN DE PANNE, M. 2007. SIMBICON: Simple Biped Locomotion Control. *ACM Trans. Graphics* 26, 3, 81.