2014 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS 2014)
September 14-18, 2014, Chicago, IL, USA

Highly Robust Running of Articulated Bipeds in Unobserved Terrain

Albert Wu and Hartmut Geyer, *Member, IEEE*

***Abstract*— Control design of running robots is often based on mapping the behavior of lower order models onto the robotic systems, and the robustness of running is largely determined by the robustness of these underlying models. However, existing implementations do not take full advantage of the stability that the low order models can provide. In particular, analysis of the theoretical spring mass model suggests leg placement policies that generate near deadbeat rejection of large, unobserved changes in ground height. Here we show in simulation that this blind robustness to rough terrain can be carried over to bipedal robots. We design a control that stably embeds the spring mass model's behavior in a planar robot model and show that resulting system rejects ground disturbances of up to 25% leg length, adapts to persistent ground slopes, and tolerates sensor noise, signal delays, and modeling errors. The results indicate that transferring control derived within the spring mass model is an effective technique for realizing highly robust running in robotic systems.**

I. Introduction

Control design of running robots can often be summarized in two distinct stages. First, a low dimensional model is used for generating a reference behavior; then, this behavior is tracked in the higher dimensional robot. Several techniques have been applied to achieve the second stage [1]–[5] As a result, the performance of running robots is largely determined by the design of the reference behavior. However, while robust locomotion in uncertain environments is a main behavioral goal, existing implementations often show limited stability in the underlying models. For instance, humanoid robots controlled by the zero moment point approach [6]–[10] generally track a single periodic reference trajectory in running [8]–[10]. Specifically, in [8], the reference trajectory is based on the low order inverted pendulum model and matches prescribed vertical motions and step locations, and it is then tracked in the joint space of the full robot. Although feedback control is applied to deviations from the reference motion, adaptations to perturbations are not considered in the lower order model. In contrast, locomotion control formulated in the hybrid zero dynamics (HZD) framework [4], [11]–[14] has been combined with local exponential stabilization of the low order model. Specifically, the authors of [14] reduce the dynamics of the full robot by introducing constraints to describe virtual compliant legs and solve for an energetically optimal limit cycle in this reduced system. They then apply discrete LQR to produce a local family of reduced order trajectories that converge to the optimal limit cycle.

This material is based on work supported in part by the US National Science Foundation (CMMI-1100232, and CPS-1239143) and the DARPA M3 program (W91CRB-11-1-0002).

A. Wu and H. Geyer are with the Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA. {apwu,hgeyer}@cmu.edu

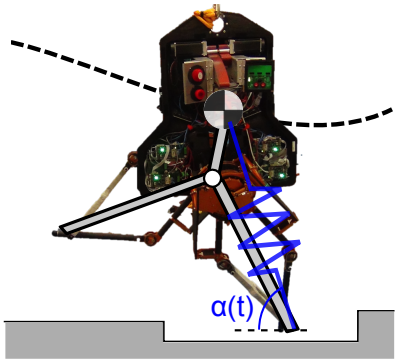

Fig. 1. Executing a swing leg trajectory $\alpha(t)$ derived from deadbeat stability of the spring mass model generates robust running in a model of a bipedal robot on unobserved rough terrain.

Thus, gait perturbations within the linear approximation of the lower order model can be gradually corrected by this controller. Finally, running robots based on the Raibert hopper [1], [15]–[17] similarly have asymptotic stability in the reference behavior. The low order Raibert hopper stabilizes the forward speed, trunk orientation, and total energy using three basic feedback laws. This stable closed loop system has been embedded in the higher order dynamics of both bipeds [18] and quadrupeds [17] by tracking a virtual leg.

In contrast to gradual error correction shown in some of the previous examples, analysis of the spring mass model (SMM) reveals a stronger level of stability available within the lower order model. In this model[1], large gait perturbations can be corrected in a single step by choosing the leg placement between apexes, thus rendering the system deadbeat stable [19]–[23]. Some recent work has made progress in simulation towards leveraging this property in control design of running robots. In [24], the authors simulate deadbeat stability for an articulated planar leg on observed rough terrain by identically reproducing the lower order behavior given perfect models of the leg and of its collision dynamics. In [25], the authors are the first to embed the 3D-SLIP as a low order model in a simulated humanoid robot and demonstrate near deadbeat tracking of target speeds as well as robustness to unmodeled impacts and pushes. Beyond these impressive results, analysis of the SMM further allows for blind robustness on rough terrain. In both 2 and 3 dimensions, the deadbeat feedback policy can be converted to a feed-forward trajectory of the swing leg that rejects disturbances from upcoming, unobserved changes in the ground height [20], [21], [23]. While this result suggests that embedding the SMM dynamics with the placement policy would facilitate robust locomotion in uncertain environments,

[1]also referred to as the spring loaded inverted pendulum (SLIP)

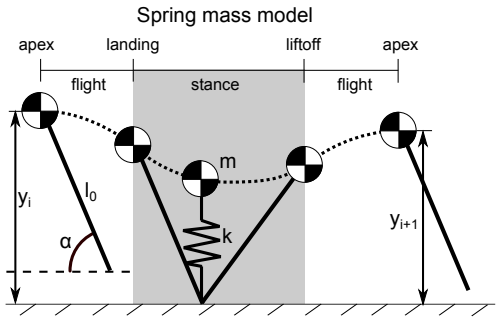

Fig. 2. The spring mass model describes running as a series of apex heights y_i, where the transitions are parameterized by the touchdown angle $\alpha_{TD,i}$. The massless leg can take on any commanded $\alpha(t)$ in flight.

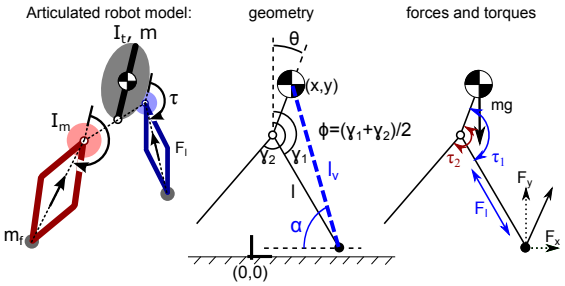

Fig. 3. The planar robot is represented by a rigid body trunk above the hips, inertias I_m and rotary actuators τ_i at each hip, linear actuators F_i in each leg, and a small mass m_f in each foot. A virtual leg can be described between the trunk COM and the stance foot. External forces F_x and F_y applied to the foot map to the robot states through a Jacobian matrix.

it has not yet been demonstrated beyond the abstract SMM.

Here we show in simulation of a planar biped robot that the blind robustness to rough terrain imparted by the deadbeat-derived swing leg trajectory is carried over to the robot system (Fig. 1). First, we review the underlying SMM and present the higher order robot system (section II). We then derive the control that maps the SMM behavior to the higher order system while stabilizing the additional dynamics of the robot (section III). Finally, we show in simulation experiments of rough terrain locomotion that the robot model rejects ground disturbances of up to 25% leg length, adapts to persistent ground slopes, and tolerates sensor noise, signal delays, and modeling errors (section IV). To the best of the authors' knowledge, this is the first time that near deadbeat stability on unobserved, very rough terrain is demonstrated on a robot model.

II. Spring Mass Model and Robot Model

A. Lower Order System: Spring Mass Model

The SMM is a low order abstraction that captures the essential behavior of locomotion on light, compliant legs [26], [27]. This model reduces the body to a point mass m following a ballistic trajectory in flight and rebounding in stance on a massless spring leg with stiffness k_0 and rest length l_0 (Fig. 2A). The massless leg has no dynamics; the landing angle α is positioned kinematically in flight. For this system, the equations of motion are

$$\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ mg \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \end{bmatrix},$$

where x and y are the COM coordinates and the ground reaction forces (GRFs) $F_x = F_y = 0$ in flight and

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = k_0(l_0 - l) \begin{bmatrix} -\cos(\alpha) \\ \sin(\alpha) \end{bmatrix}$$

in stance. The leg angle α and leg length l are functions of the COM coordinates assuming a stationary foot in stance.

Under these dynamics, it has been shown that deadbeat gait stability can be achieved by choosing the leg placement α in flight. Since the SMM is conservative, each flight phase is fully characterized by the apex height y_i, and the trajectory through the ensuing stance and flight phases is then fully determined given the touchdown angle $\alpha_{TD,i}$. Thus, SMM running is described as a series of apex heights for which the transitions depend on the landing angle: $y_{i+1} = f|_{y_i}(\alpha_{TD,i})$. To track a desired gait described by a target height y^*, from any initial condition y_i, deadbeat stability is realized by landing with angle

$$\alpha^*_{TD,i} = f^{-1}|_{y_i}(y^*) \tag{1}$$

at some specific time t after apex.

Blind tolerance to rough terrain is embedded by approximating upcoming terrain changes as unobserved perturbations to the previous apex height y_i (the approximation is inexact due to energetic differences) [20], [23]. Numerically solving Eq. 1 off-line for all possible values of y_i yields solution pairs $(\alpha^*_{TD,i}, t)$ describing touchdown angles coupled with their respective times of landing. Sequencing the solutions by time gives a touchdown angle profile $\alpha^*(t)$ for the swing leg. In effect, whenever touchdown on unobserved terrain does occur, the initial condition y_i is implicitly inferred and $\alpha^*(t)$ provides the correct landing condition.

B. Higher Order Robot: ATRIAS

ATRIAS (Fig. 1), a bipedal robot developed by the Dynamic Robotics Laboratory at Oregon State University [28], is well suited to embed the behavior of the SMM. It has very lightweight carbon-fiber legs and point feet. Each leg is composed of a four-bar mechanism actuated through leaf springs by a pair of motors at the hip, which act as differential drives to generate leg rotation as well as leg extension and shortening. In effect, the robot's total mass is concentrated on the trunk, resembling a point mass of about 60kg rebounding on massless and compliant legs about 1m in length. In contrast to the mathematical abstractions made in the SMM, ATRIAS has a trunk that needs to be balanced, legs that need to generate actual swing motions with ground foot clearance, and feet that produce actual contact dynamics with slipping and energy dissipation. In addition, the robot is actuated by geared motors (50:1 harmonic drives), which have torque saturation and rotational inertias.

We simulate a planar robot model that captures the essential characteristics of this system (Fig. 3). It has a trunk of mass m and inertia I_t about the center of mass, which is located distance d away from the hip joint. The effect of each four-bar leg mechanism is represented by two components: a

linear actuator outputting force F_i that extends the leg with a small foot point mass m_f, and a rotary actuator with limited torque $|\tau_i| < \tau_{\max}$ coupled to an inertia I_m at the hip.

Whenever a leg touches the ground, a dynamic contact model actuates the foot. In the vertical direction, the ground behaves like a nonlinear spring and damper, generating the force $GRF_y = k_g \Delta y_g (1 - \dot{y}_g / v_{\max})$. In the horizontal direction, it generates either a similarly computed nonlinear spring and damper stiction force, or a sliding force that opposes the motion in proportion to GRF_y. Details of the contact model are described in [29], and it realistically describes unilateral forcing, dissipative collisions, and both static and sliding friction.

These dynamics are summarily described in the equations of motion

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}\tau + \mathbf{J}^T\mathbf{F} \tag{2}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{q}$ is the coordinate vector, vector $\mathbf{h}$ accounts for Coriolis, centrifugal, and gravitational forces, the selection matrix $\mathbf{S}$ assigns control inputs τ to $\mathbf{q}$, and the Jacobian $\mathbf{J}$ maps external forces $\mathbf{F}$ to $\mathbf{q}$. Here, $\mathbf{q}$ consists of the coordinates x, y and θ describing the global position and orientation of the trunk, along with γ_i and l_i describing the global orientation and length of each leg (Fig. 3). Control vector τ is composed of the motor torque τ_i and linear force F_i of each leg, and vector $\mathbf{F}$ holds the components of the external force applied at each foot. The modeling of these GRFs is described above, and they cannot be expressed as functions of just the robot state and applied actuation.

III. Mapping Control Design

A. Simplifying Assumptions

In deriving the control, we apply a few assumptions to simplify the dynamics and reduce the complexity of the design process. Our simulation results (section IV) confirm that the robot runs robustly despite the errors induced by these assumptions.

First, because $m_f/m \ll 1$, we ignore the inertial effects of the masses at the feet, thereby eliminating small coupling terms in $\mathbf{M}$ and $\mathbf{h}$ and collapsing the COM onto the trunk coordinate (x, y). Setting $m_f = 0$ would also imply $F_i = 0$ and remove the dynamics of the length l_i for any leg not in contact with the ground. Instead, the leg force of a swing leg is computed separately using a PD feedback loop that tracks a target leg length. In stance, the simplified dynamics of the remaining states is given by the equation

$$\begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & I_t & 0 & 0 & 0 \\ 0 & 0 & 0 & I_m & 0 & 0 \\ 0 & 0 & 0 & 0 & I_m & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \\ \ddot{\gamma}_1 \\ \ddot{\gamma}_2 \\ \ddot{l} \end{bmatrix} + \begin{bmatrix} 0 \\ mg \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ F_1 \end{bmatrix} + \mathbf{J}^T \begin{bmatrix} F_x \\ F_y \end{bmatrix} \tag{3}$$

where index 1 denotes the stance leg and 2 denotes the swing leg, l is the length of the stance leg, $F_{x,y}$ are the horizontal and vertical GRFs (Fig. 3) at the stance leg, and the Jacobian $\mathbf{J}$ is given by

$$\mathbf{J} = \begin{bmatrix} J_x \\ J_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & -d\cos\theta & l\cos\gamma_1 & 0 & \sin\gamma_1 \\ 0 & 1 & d\sin\theta & -l\sin\gamma_1 & 0 & \cos\gamma_1 \end{bmatrix}. \tag{4}$$

In swing, no external force acts on either leg, so (3) further reduces to

$$\begin{bmatrix} m & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 \\ 0 & 0 & I_t & 0 & 0 \\ 0 & 0 & 0 & I_m & 0 \\ 0 & 0 & 0 & 0 & I_m \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \\ \ddot{\gamma}_1 \\ \ddot{\gamma}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ mg \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}, \tag{5}$$

leaving only the angular states and a ballistic COM trajectory.

The other assumption in designing the control replaces the dynamic contact model governing the GRFs with a standard constraint; the ground provides the exact force which prevents motion of the foot. Written as a function of coordinates $\mathbf{q}$ (the remaining states in Eq. 3), this requirement of $\dot{x}_{\text{foot}} = 0$ and $\dot{y}_{\text{foot}} = 0$ implies the acceleration constraints

$$\mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}} = 0$$

where $\mathbf{J}$ is the same Jacobian in Eq. 4. Combining the floating base equations of motion and constraints into a single system of equations yields

$$\begin{bmatrix} \mathbf{M}_{6\times 6} \\ \mathbf{J}_{2\times 6} \end{bmatrix} \ddot{\mathbf{q}} + \begin{bmatrix} \mathbf{h}_{6\times 1} \\ (\dot{\mathbf{J}}\dot{\mathbf{q}})_{2\times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{6\times 3} \\ \mathbf{0}_{2\times 3} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ F_1 \end{bmatrix} + \begin{bmatrix} (\mathbf{J}^T)_{6\times 2} \\ \mathbf{0}_{2\times 2} \end{bmatrix} \begin{bmatrix} F_x \\ F_y \end{bmatrix}, \tag{6}$$

which relates robot accelerations, actuator inputs, and reaction forces.

In section III-E, we discuss an alternative assumption for when a static contact point is invalid.

B. Control Goals

In the SMM, the leg placement policy $\alpha(t)^*$ provides blind, deadbeat tracking of a specified gait (parameterized by apex height and forward velocity $[y_a^*, \dot{x}_a^*]$, where $\dot{x}_a^*$ resolves the energy of the gait) in the presence of large ground disturbances. The higher order robot recovers the same performance if it tracks the equivalent leg placement policy in flight and generates GRFs matching a virtual spring through the COM in stance. Beyond this direct embedding of the SMM, there are three other tasks that must be addressed in the higher order system. First, the trunk pitch θ must be stabilized, and the total angular momentum $H = I_t\dot{\theta} + I_m\dot{\gamma}_1 + I_m\dot{\gamma}_2$ about the COM should be 0 on average; otherwise the system would have an eventually unstable pitch rate. Since the GRFs of the SMM are directed through the COM such that H is nominally constant, here we target $H(t) = 0$ for all time. Second, the translational energy E_{trans} (defined without rotational components) should match the system energy E^* that parameterizes the SMM. Both E_{trans} and H are constant in flight (see Eq. 5); they can only be regulated in stance. Finally, the robot should be driven away from problematic configurations. For example, if the trailing leg starts too far back (γ_2 is too large) in stance, swinging it forward quickly to track $\alpha^*(t)$ in flight can be difficult. Mirroring the two legs through $\gamma_2 = 2\pi - \gamma_1$ solves this problem by targeting symmetric configurations. Furthermore, coupled with driving the trunk to a constant reference pitch θ^*, mirroring angular velocities $\dot{\gamma}_2 = -\dot{\gamma}_1$ also encodes zeroing the angular momentum H.

Therefore, on a functional level, we implement controls with the following specific goals:

1) Track SMM-derived landing angle $\alpha^*(t)$ in flight.
2) Produce GRFs matching the SMM in stance.
3) Maintain trunk pitch θ^* throughout flight and stance.
4) Stabilize system energy to E^* in stance.
5) Drive the average hip angle $\phi \equiv \frac{\gamma_1+\gamma_2}{2}$ to $\phi^* = \pi$, which enforces mirrored leg angles.

The remainder of this section details how we design the control to achieve these goals during flight, stance, and sliding transitions. The methods we use include optimal control, inverse dynamics, and feedback linearization, and are similar to the methods used by others for mapping control in robotic systems.

C. Flight

In flight, the center of mass moves on a ballistic trajectory (Eq. 5), and the control focuses on the angular motions of the trunk and legs. As listed above, the control goal is to track the SMM-derived angle $\alpha^*(t)$, the trunk pitch θ^*, and the average hip angle ϕ^*. Since angular momentum is conserved in flight, not all these goals can be met simultaneously. We use linear quadratic regulation (LQR) to optimally trade off between these goals. Meanwhile, the control of the leg lengths is independent and addressed in Section III-F.

The angular dynamics are linear. Expressed in state space form $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$, we obtain

$$\dot{\mathbf{x}} = \begin{bmatrix} 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 & 0 & 0 & -\frac{1}{I_t} & \frac{1}{I_m} & 0 \\ 0 & 0 & 0 & -\frac{1}{I_t} & 0 & \frac{1}{I_m} \end{bmatrix}^T \mathbf{u}$$

with $\mathbf{x} = [\theta\ \gamma_1\ \gamma_2\ \dot{\theta}\ \dot{\gamma}_1\ \dot{\gamma}_2]^T$ and $\mathbf{u} = [\tau_1\ \tau_2]^T$. The controllability matrix $[\mathbf{A}\ \mathbf{AB}\ \mathbf{A^2B}]$ is rank deficient by 2; the system is uncontrollable, corresponding to the fact that no external torque acts on the COM and the total angular momentum H is conserved. Thus, the target behavior defined with respect to all three segments may not be achievable. Instead, the controller must track the targets as closely as the dynamics allow. The combined tracking tasks are expressed in a quadratic objective function

$$\mathcal{J} = \int_0^\infty (\mathbf{x}-\mathbf{x}^*)^T\mathbf{Q}(\mathbf{x}-\mathbf{x}^*) + (\mathbf{u}-\mathbf{u}^*)^T\mathbf{R}(\mathbf{u}-\mathbf{u}^*)dt$$

where the tracking targets $\gamma_1^*(t)$, θ^* and ϕ^* feature in $\mathbf{x}^*$, and the hip angle $\gamma_1^*(t) = \frac{\pi}{2}+\alpha^*(t)-\sin^{-1}(d\sin(\gamma_1-\theta)/l_v)$ is defined by the target angle $\alpha^*(t)$ of the virtual leg (Fig. 3). LQR solves for the optimal feedback policy $\mathbf{u} = \mathbf{u}^* - \mathbf{K}(\mathbf{x}-\mathbf{x}^*)$ that minimizes $\mathcal{J}$. However, the uncontrollable modes cause this integral to grow unbounded.

To circumvent this problem, we rewrite the uncontrollable modes in terms of the momentum H, reducing the explicit states to a controllable subsystem. In particular, we substitute $\dot{\gamma}_2 = \frac{1}{I_m}(H - I_t\dot{\theta} - I_m\dot{\gamma}_1)$ and integrate this expression to obtain $\gamma_2 = f(H, t, \mathbf{x}_0, \mathbf{x}_r)$ as a function of H, time, the initial conditions $\mathbf{x}_0$, and the reduced states $\mathbf{x}_r \equiv [\theta\ \gamma_1\ \dot{\theta}\ \dot{\gamma}_1]$. The dynamics of this controllable subsystem are

$$\dot{\mathbf{x}}_r = \underbrace{\begin{bmatrix} 0_{2\times2} & I_{2\times2} \\ 0_{2\times2} & 0_{2\times2} \end{bmatrix}}_{\mathbf{A}_r}\mathbf{x}_r + \underbrace{\begin{bmatrix} 0 & 0 & -\frac{1}{I_t} & \frac{1}{I_m} \\ 0 & 0 & -\frac{1}{I_t} & 0 \end{bmatrix}^T}_{\mathbf{B}_r}\mathbf{u}$$

and the objective function reduces to

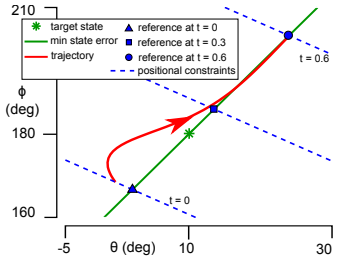

Fig. 4. Applying LQR after accounting for uncontrollable angular momentum in a 2 dimensional system with states $[\theta\ \phi\ \dot{\theta}\ \dot{\phi}]$. Integrating $H = I_t\dot{\theta} + 2I_m\dot{\phi}$ yields a time-dependent linear constraint between θ and ϕ. The reachable sets $[\theta\ \phi](t)$ satisfying these constraints are shown as dotted blue lines for t = (0s, 0.3s, 0.6s). The original target $[\theta^*\ \phi^*]$ (green star) is only attainable at one specific time (t $\approx$ 0.2s). At all other times, there is a distinct state (blue markers) within the reachable set that minimizes the state error. Targeting these states in series produces a trajectory (green line) of lowest-cost feasible states. The re-parameterized problem (explicit dependence on $[\phi\ \dot{\phi}]$ removed) is controllable, and LQR w.r.t the moving references $\theta^\dagger(t)$ produces the optimal red trajectory.

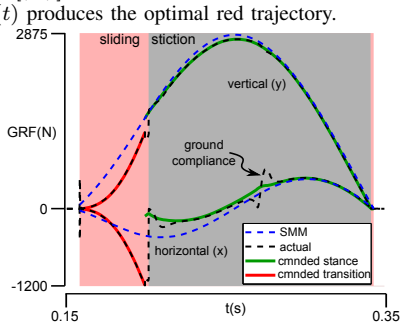

Fig. 5. Vertical and horizontal GRFs during the first stance phase. The commanded GRFs from the stance controller (solid green) differ slightly from the SMM forces (dashed blue) due to the change in stiffness for energy regulation and to the feedback torques for correcting the trunk and swing leg orientations and velocities. The simulated forces (dashed black) do not exactly match the commanded forces because the assumptions used for deriving the controller differ from the simulated dynamics. At the beginning of stance, the contact is not stationary, and the transition controller (solid red) accelerates the foot while gradually matching the vertical loading of the SMM model. The controller's switching to stance control (section III-G) does not exactly match the simulation's transition between kinetic friction (shaded pink) and stiction (shaded gray), but this difference does not result in instability.

$$\mathcal{J} = \underbrace{\int_0^\infty (\mathbf{x}_r-\mathbf{x}_r^\dagger)^T\mathbf{Q}_r(\mathbf{x}_r-\mathbf{x}_r^\dagger) + (\mathbf{u}-\mathbf{u}^\dagger)^T\mathbf{R}(\mathbf{u}-\mathbf{u}^\dagger)}_{\mathcal{J}_r} + g(H,t,\mathbf{x}_0)dt$$

where $g(H, t, \mathbf{x}_0)$ is the inevitable error that accrues as a result of the fixed angular momentum. With no dependence on $\mathbf{u}$ or $\mathbf{x}_r$, we drop g from the objective function and solve LQR for the matrices $(\mathbf{A}_r,\ \mathbf{B}_r,\ \mathbf{Q}_r, \mathbf{R})$ to find the optimal control $\mathbf{u} = \mathbf{u}^\dagger - \mathbf{K}(\mathbf{x}_r - \mathbf{x}_r^\dagger)$ that minimizes $\mathcal{J}_r$, the part of $\mathcal{J}$ that does not grow unbounded. (The algebra for solving $\mathbf{Q}_r$, $\mathbf{x}_r^\dagger$, and $\mathbf{u}^\dagger$ is omitted here for brevity.) The resulting control uses a linear gain matrix $\mathbf{K}$ to track a compensated reference $\mathbf{x}_r^\dagger(t)$ that is optimal with respect to the original target state trajectory $\mathbf{x}^*$, subject to the constant angular momentum. As an example, Fig. 4 visualizes the resulting tracking behavior for a simpler, two-segment system ($[\theta\ \phi]$).

D. Stance

The control goals in stance consist of tracking SMM ground reaction forces GRF^*, maintaining trunk pitch θ^*, stabilizing system energy E^*, and enforcing the average

hip angle ϕ^*. Because energy regulation and producing energetically conservative SMM behavior compete with each other as control goals, we use a cascaded control design. An outer loop regulates system energy by modulating the virtual spring stiffness, and an inner loop uses inverse dynamics control [30] to generate leg forces that are similar to the SMM while stabilizing pitch and average hip angle.

Outer loop - energy regulation: The stance dynamics of the SMM are characterized by ground reaction forces $GRF^* = k_0 \Delta l \mathbf{l}_v / |\mathbf{l}_v|$ where k_0 is the virtual spring stiffness and $\mathbf{l}_v$ is the virtual leg pointing from the foot to the COM. We alter this target force profile to regulate the translational system energy $E_{\text{trans}} = \frac{m}{2}(\dot{x}^2 + \dot{y}^2) + mgy$, which we estimate from the robot configuration at lift-off. Based on the error $\Delta E = E_{\text{trans}} - E^*$, the target GRFs are modified assuming a spring-leg behavior with the nonlinear stiffness $k = k_0 + c\Delta E \dot{l}_v$. The virtual spring force F_v produces the net work

$$W = \int_{t_0}^{t_f} F_v \cdot \dot{l}_v dt = -\int_C k_0 \Delta l_v dl_v - c\Delta E \int_{t_0}^{t_f} \Delta l_v \dot{l}_v^2 dt.$$

The first term integrates to zero (linear spring component) while the second term always opposes the error ΔE. The modification preserves a GRF target profile

$$\begin{bmatrix} F_{v,x} \\ F_{v,y} \end{bmatrix}^* = -k\Delta l_v \begin{bmatrix} (d\sin(\theta) - l\sin(\gamma_1))/l_v \\ (d\cos(\theta) - l\cos(\gamma_1))/l_v \end{bmatrix} \quad (7)$$

that is similar to the SMM, with the correction gain c balancing between energy correction and profile similarity.

Inner loop - inverse dynamics and positional feedback: Next we compute the actuator commands for producing these target GRFs, and then add a feedback controller for positional tracking. To match nominal values of F_x, F_y, and $\ddot{\theta}$, we re-arrange the full equations of motion (Eq. 6) as

$$\underbrace{\begin{bmatrix} -\mathbf{S}_{6\times3} & \mathbf{M}'_{6\times5} \\ \mathbf{0}_{2\times3} & \mathbf{J}'_{2\times5} \end{bmatrix}}_{\mathbf{P}} \mathbf{z} = \begin{bmatrix} (\mathbf{J}^T)_{6\times2} & -\mathbf{M}^\circ_{6\times1} \\ \mathbf{0}_{2\times2} & -\mathbf{J}^\circ_{2\times1} \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ \ddot{\theta} \end{bmatrix} - \begin{bmatrix} \mathbf{h}_{6\times1} \\ (\dot{\mathbf{J}}\dot{\mathbf{q}})_{2\times1} \end{bmatrix} \quad (8)$$

where vector $\mathbf{z} = [\tau_1 \ \tau_2 \ F_1, \ \ddot{x} \ \ddot{y} \ \ddot{\gamma}_1 \ \ddot{\gamma}_2 \ \ddot{l}]$, vectors $\mathbf{M}^\circ$ and $\mathbf{J}^\circ$ select the 3rd column (corresponding to θ in $\mathbf{q}$) of $\mathbf{M}$ and $\mathbf{J}$, leaving $\mathbf{M}'$ and $\mathbf{J}'$ as the remaining 5 columns from each. The matrix $\mathbf{P}$ is invertible, allowing us to directly solve for the actuation and resulting accelerations in $\mathbf{z}$ for a specified target $[F_x \ F_y \ \ddot{\theta}]$. In line with inverse dynamics control [30], we substitute the virtual spring forces $[\ F^*_{v,x}, \ F^*_{v,y}, 0]$ from Eq. 7 for $[F_x, F_y, \ddot{\theta}]$ in Eq. 8 and extract the first 3 elements of $\mathbf{z}$ to find the feed-forward actuation $[\tau_{1,\text{ff}}, \ \tau_{2,\text{ff}}, \ F_{1,\text{ff}}]$ that produces the nominal behavior. Finally, we apply $\tau_i = \tau_{i,\text{ff}} + \tau_{i,\text{fb}}$ with the feedback torques

$$\tau_{2,\text{fb}} = K_{p\phi}(2\pi - \gamma_1 - \gamma_2) - K_{d\phi}(\dot{\gamma}_1 + \dot{\gamma}_2),$$
$$\tau_{1,\text{fb}} = -\tau_{2,\text{fb}} + K_{p\theta}(\theta^* - \theta) - K_{d\theta}(\dot{\theta}),$$

to drive the average hip angle ϕ and the trunk angle θ to stationary references π and θ^* with second order error dynamics. Leg force is not used in the feedback, so we directly apply $F_1 = F_{1,\text{ff}}$. These actuator outputs nominally generate the green GRFs in the stiction phase of stance in Fig. 5.

E. Sliding

The constraint $\dot{x}_{\text{foot}} = 0$ is not always a good assumption. In particular, upon touchdown, the relative foot speed is largely determined by the tracking of the landing policy $\alpha^*(t)$ and is entrenched in the hip rotating with inertia I_m, but it generally does not exactly oppose the center of mass speed $\dot{x}$. The previous stance control is not ideal for this transition phase because the inverse dynamics in Eq. 8 assumes F_x maintains $\dot{x}_{\text{foot}} = \ddot{x}_{\text{foot}} = 0$, firmly opposing applied actuation. Even with a rough estimate for the coefficient μ of sliding friction, $|F_x| = \mu F_y$ is a more appropriate assumption. For the running conditions we simulate, we safely replace the absolute value with a negative sign.

This transition phase does not exist in the abstract SMM dynamics that we wish to embed. Furthermore, the orientation of the GRF vector is governed by μ which can be hard to estimate, allowing the controller very limited authority for regulating momentum or joint configuration. Therefore, the goal during this transient phase is to quickly and smoothly shift to the subsequent stance phase where $\dot{x}_{\text{foot}} \approx 0$.

Thus, during sliding, we apply maximum torque τ_1 while mimicking the nominal stance behavior. Equations 7 and 8 of the stance controller yield the vertical reaction force $F^*_{v,y}$ and leg acceleration $\ddot{l}_{\text{nom}}$ (last element of $\mathbf{z}$) for the target spring-like behavior. Reproducing $\ddot{l}^*_{\text{slide}} = \ddot{l}_{\text{nom}}$ during the transition largely preserves the virtual leg geometry of the SMM, and scaling the target vertical force $F^*_{y,\text{slide}} = \max(0, \dot{x}_r/\dot{x}_a)^2 F^*_{y,v}$ based on the relative motion of the foot avoids prematurely loading the leg. The scaling term compares the relative velocity between the COM and the foot $\dot{x}_r$ to the estimated apex speed $\dot{x}_a$ (estimated at the end of the previous stance phase).

The equations of motion (Eq. 3) combined with the sliding friction assumption and a vertically constrained point of contact yields the system of equations

$$\begin{bmatrix} \mathbf{M}_{6\times6} \\ \mathbf{J}_{y,1\times6} \\ \mathbf{0}_{1\times6} \end{bmatrix} \ddot{\mathbf{q}} + \begin{bmatrix} \mathbf{h}_{6\times1} \\ (\dot{\mathbf{J}}_{\mathbf{y}}\dot{\mathbf{q}})_{1\times1} \\ 0_{1\times1} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{6\times3} \\ \mathbf{0}_{1\times3} \\ \mathbf{0}_{1\times3} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ F_1 \end{bmatrix} + \begin{bmatrix} (\mathbf{J}^T)_{6\times2} \\ \mathbf{0}_{1\times2} \\ [1 \ \ \mu] \end{bmatrix} \begin{bmatrix} F_x \\ F_y \end{bmatrix}.$$

Finally, to solve for the actuation to produce the target behavior, we re-arrange the system of equations as

$$\begin{bmatrix} -\mathbf{S}'_{6\times2} & \mathbf{M}'_{6\times5} & -(\mathbf{J}^T_x)_{6\times1} \\ \mathbf{0}_{1\times2} & \mathbf{J}'_{y,1\times5} & 0_{1\times1} \\ \mathbf{0}_{1\times2} & \mathbf{0}_{1\times5} & -1 \end{bmatrix} \mathbf{w} =$$
$$\begin{bmatrix} \mathbf{S}^\circ_{6\times1} & (\mathbf{J}^T_y)_{6\times1} & -\mathbf{M}^\circ_{6\times1} \\ 0_{1\times1} & 0_{1\times1} & -J^\circ_{y,1\times1} \\ 0_{1\times1} & \mu & 0_{1\times1} \end{bmatrix} \begin{bmatrix} \tau_1 = \tau_{\max} \\ F_y = F^*_{y,\text{slide}} \\ \ddot{l} = \ddot{l}^*_{\text{slide}} \end{bmatrix} - \begin{bmatrix} \mathbf{h}_{6\times1} \\ (\dot{\mathbf{J}}_{\mathbf{y}}\dot{\mathbf{q}})_{1\times1} \\ 0_{1\times1} \end{bmatrix}$$

where the first matrix is invertible and the first two elements of $\mathbf{w} = [\tau_2 \ F_1, \ \ddot{x} \ \ddot{y} \ \ddot{\theta} \ \ddot{\gamma}_1 \ \ddot{\gamma}_2, \ F_x]$ give the remaining actuator commands. Similar to Eq. 8, $\mathbf{M}^\circ$ and J°_y select the 2nd column (for θ in $\mathbf{q}$) of $\mathbf{M}$ and $\mathbf{J}_y$ while vector $\mathbf{S}^\circ$ selects the 1st column of $\mathbf{S}$ (for τ_1), respectively leaving 5 columns in $\mathbf{M}'$ and $\mathbf{J}'_y$and 2 columns in $\mathbf{S}'$. These actuator outputs nominally generate the red GRFs in the sliding phase of stance in Figure 5.

F. Foot ground clearance

Omitting the foot mass removes the length of the swing leg from the equations of motion (Eqs. 3 and 5). During swing, the leg force is computed using a PD feedback loop that tracks a target leg length. For the trailing leg, a target length is commanded such that the foot height scales (50%) with the estimated COM height. After apex, the leading leg begins to track the nominal landing length. This change of goals scales linearly with the landing angle error $|\gamma_1^* - \gamma_1|$.

G. Switching between modes

Assumed contact sensors in the foot trigger the controller to switch from flight to sliding transition. The control mode then changes to stance when two conditions are met: the relative COM velocity $\dot{x}_r$ matches the estimated apex velocity $\dot{x}_a$ within 10%, and modeling the stance controller's commands using the slipping dynamics produces an acceleration $\ddot{x}_r$ that reduces the estimated speed difference. This prevents the stance controller from prolonging slipping. Note that explicit detection of the transition between actual sliding and stiction is not required by the control. Finally, the control switches back to flight when the distance l_v exceeds the virtual rest length of the SMM.

IV. Simulation Results

We test the performance of the derived control on the simulated system in Section II-B without the assumptions made in Section III-A for deriving the control. Other disturbances are introduced through rough terrain, sensor noise, delays, and mis-modeled parameters. The controller robustly stabilizes the gait ($[y_a, \dot{x}_a]$) to the target [1.05m, 5m/s] and while stabilizing the trunk orientation θ to a reference lean of $\theta^* = 10°$, despite these differences and disturbances (Fig. 6 shows the simulated robot running over unobserved rough terrain). Tracking the tasks defined at the beginning of section III successfully achieves the deadbeat gait stability derived from the abstract SMM on the simulated articulated robot.

A. Performance on flat ground

On flat ground, the simulated robot stabilizes an initial trunk perturbation of $10°$ in about two steps. Figs. 7A,B,C show the performance with respect to the original performance objective of tracking the gait parameterized by apex heights and speeds $[y_a, \dot{x}_a]$, while holding the trunk at θ^*. The gait behavior is achieved through matching the landing condition $\alpha^*(t)$ (dashed green in Fig. 7D) in flight (unshaded) while producing SMM-consistent GRFs (Fig.5) during stance (shaded pink and gray). The discontinuities in α (black in Fig. 7D) come from switching the definition of the landing leg. In flight, the controller computes and tracks adjusted target angles $[\theta^\dagger, \phi^\dagger]$ (dashed green in Figs. 7C,E) that are not necessarily $[\theta^* = 10°,\ \phi^* = 180°]$ but are consistent with the fixed angular momentum H (Fig. 7F). In addition to regulating energy, the stance control drives H back to 0 through small changes in the GRF profile when the foot is stationary (shaded gray). When the foot is slipping

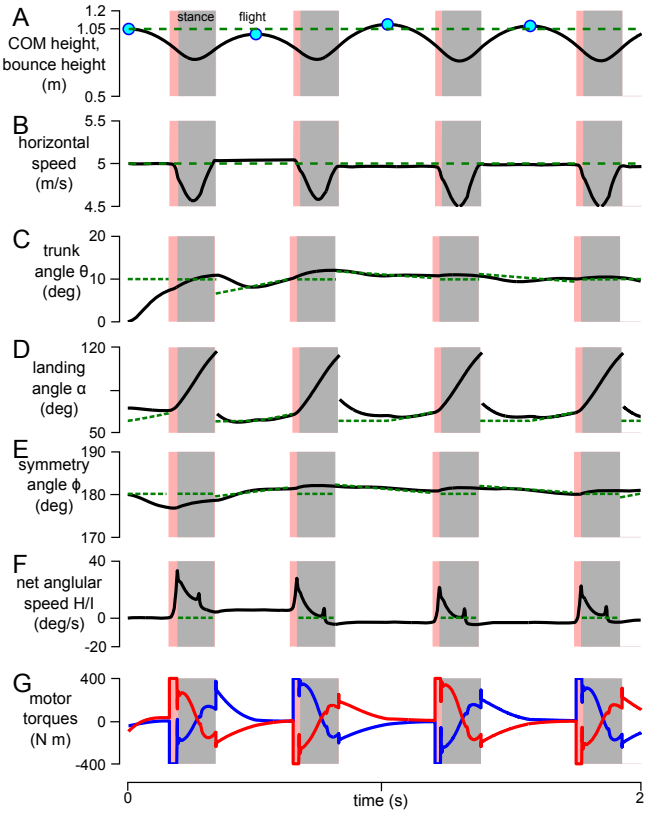

Fig. 7. First 2 seconds of simulated running on flat terrain. White background for flight, pink for transition, and gray for stance. Target values in dashed green and only shown in the phases in which the states are actively controlled. See section IV-A for detailed description.

(shaded red), the motors saturate (Fig. 7G) to oppose the motion of the foot.

The resulting apex heights (light blue circles in Fig. 7A) and horizontal speeds (Fig. 7B) match the desired values with respective steady state errors of 1.8cm and 0.8cm/s. These errors are comparable to the errors in a pure SMM simulation on flat terrain (-1.1cm, 2.3cm/s) induced by approximating $\alpha(t)$ with the same polynomial used for the articulated robot controller.

B. Robustness to unobserved rough terrain

On rough terrain with random steps every 3m that have height changes up to $\pm$25cm, the robot model does not fall and performs similarly to the pure SMM model (Figs. 6 and 8).

By design, the SMM leg placement policy targets an apex "bounce height" (blue circles in Figs. 7A and 8) measured with respect to the ground level at the previous stance phase, as opposed to the absolute apex height within the world frame (black lines). The terrain variation changes the effective system energy [23], which introduces small errors (s.d. 1.8cm and 18.6cm/s) in the SMM bounce height and speed tracking. Using joint sensor data at the end of stance, the robot model can only estimate the relative energy with respect to the ground at stance. This quantity includes the energy introduced by variations in the terrain as well as energy dissipation due to robot motion. To compare the

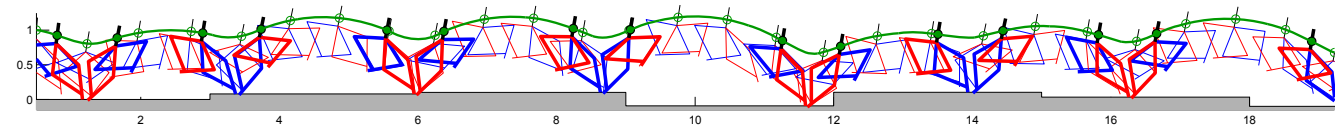

Fig. 6. First 4 seconds of simulation over rough terrain.

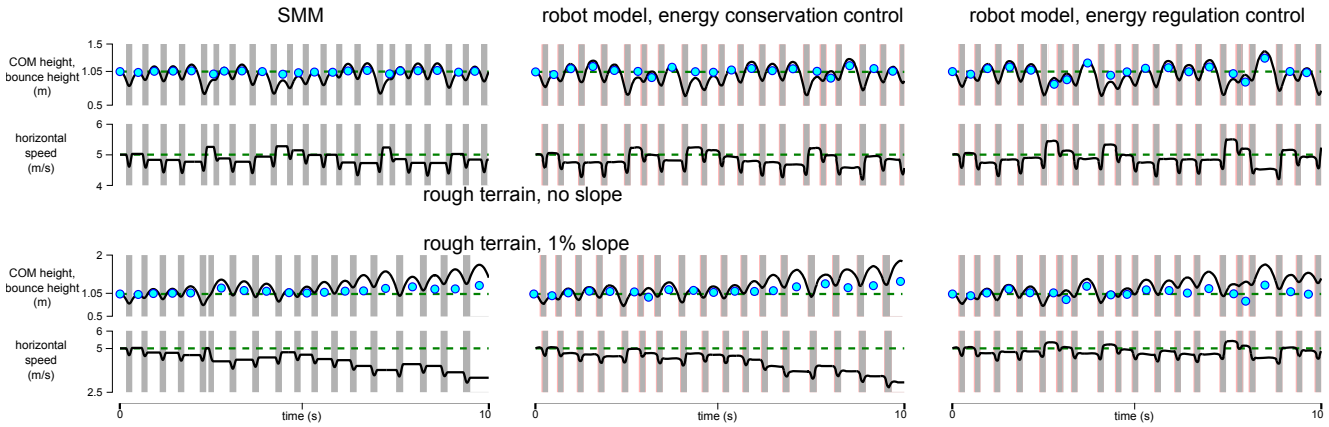

Fig. 8. Performance of the pure SMM and of two versions of the robot model (see Section IV-B for details) on rough terrain with maximum step changes of 25cm for ground that is flat on average (top row) and sloped upward on average (bottom row). In all cases, the leg placement targets a bounce height of 1.05cm and a horizontal speed of 5m/s. The difference between the global heights (black lines) at apex and the bounce heights (light blue circles) is the terrain height at stance measured in the global frame. The performance of the full robot is comparable to that of the underlying model.

tracking performance to that of the SMM, we implement two versions of energy regulation in the robot model. The first version only accounts for the dissipative losses (energy conservation control) by using truth values for y_{foot}. Albeit unrealistic for actual robots, it enables a direct performance comparison to the SMM (first two columns in Fig. 8). The second version regulates the total energy difference estimated from joint sensor data (right column, energy regulation control) by assuming $y_{\text{foot}} = 0$. The first version has slightly better bounce height and speed tracking (s.d. 5.3cm and 20.2cm/s) than the second one (s.d. 10.4cm and 25.5cm/s), which works harder to regulate the small energetic deviations between steps instead of maintaining the prescribed gait.

Because it considers the most current ground height, the more realistic energy regulation control generalizes to sloped terrain without further modification. With a persistent 1% slope added to the rough terrain, the gait tracking of this controller is unaffected (s.d. 10.2cm and 26.4cm/s), while the pure SMM model and the robot model with energy conservation control eventually fail by losing forward velocity as the slope consumes effective locomotive energy.

The tracking of the trunk angle, the landing angle, average hip angle, and angular momentum in all instances of the robot model are very similar to the flat ground simulation shown in Fig. 7 and are omitted here.

C. Error sensitivity

Finally, we examine the sensitivity of the controller to sensor noise, delays, and modeling errors (using the energy conservation control version). With uniform noise (sampled every 3ms) distributed between $\pm 3^\circ$, $\pm 3^\circ$/s, $\pm$3cm, and $\pm$3cm/s for all state sensors, the model stably runs over the same rough terrain with a wider spread of apex heights and speeds (s.d. 10.1cm and 26.1cm/s). In addition, the controller tolerates a 10ms delay over rough terrain, although it runs significantly lower and faster (average errors of -7.2cm and +71.9cm/s). This overshoot may be caused by the delay of force generation, which reduces braking in the first half of stance and increases acceleration in the second half at lower speeds. Computing the inverse dynamics with inertias or a coefficient of friction with $\pm 50\%$ errors has very little effect on orientation tracking, momentum regulation, and gait tracking over rough terrain. Miscalculated mass or spring stiffness have larger effects as they produce errors in foot placement policy. For instance, $\pm 20\%$ errors in mass or stiffness cause average tracking errors of about [$\pm$14cm, $\mp$17cm/s] and [$\mp$6cm, $\pm$5cm/s], respectively. Despite these errors, the system keeps running on both flat and rough terrain.

V. Conclusion

In this work, we design a control for a higher order robot model by embedding the deadbeat stabilized spring mass system. Specifically, we incorporate a deadbeat-derived leg placement policy that renders the running robot robust to very rough, unobserved terrain. To the best of our knowledge, this level of stability has not been demonstrated either on hardware or in simulation. Furthermore, the tolerance shown by the simulated robot to a wide array of disturbances and uncertainties strongly suggests that building the stability of the SMM into the underlying reference behavior can be a practical and powerful technique for controlling real running systems. Our current research addresses this step. We are working on transferring the control developed here to a

Fig. 9. CMU's copy of ATRIAS [28], currently mounted on a boom to validate planar running using the deadbeat leg placement policy from the SMM.

specific bipedal robot (Fig. 9). In addition, we are extending this control framework to 3D systems. As an initial result, we have extended the leg placement policy to produce deadbeat running with steering over unobserved rough terrain [23]. Our current focus in this area is to map this behavior onto the 3D dynamics of the higher order robot.

REFERENCES

[1] M. Raibert, *Legged Robots That Balance*. MIT press, Cambridge, 1986.

[2] T. Mita and T. Ikeda, "Proposal of a variable constraint control for sms with application to a running and jumping quadruped," in *Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on*, vol. 3, pp. 140–145, IEEE, 1999.

[3] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *Proceedings of the 6th IEEE-rAS Intl Conf on Humanoid Robots*, 2006.

[4] E. Westervelt and J. Grizzle, *Feedback Control of Dynamic Bipedal Robot Locomotion*. Control and Automation Series, CRC PressINC, 2007.

[5] L. Righetti, J. Buchli, M. Mistry, and S. Schaal, "Inverse dynamics control of floating-base robots with external constraints: A unified view," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 1085–1090, IEEE, 2011.

[6] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Resolved momentum control: humanoid motion planning based on the linear and angular momentum," in *Intelligent Robots and Systems*, pp. 1644–1650, 2003.

[7] M. Hirose and K. Ogawa, "Honda humanoid robots development," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1850, pp. 11–19, 2007.

[8] S. Kajita, T. Nagasaki, K. Kaneko, and H. Hirukawa, "Zmp-based biped running control," *Robotics & Automation Magazine, IEEE*, vol. 14, no. 2, pp. 63–72, 2007.

[9] T. Takenaka, T. Matsumoto, T. Yoshiike, and S. Shirokura, "Real time motion generation and control for biped robot 2nd report: Running gait pattern generation," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 1092–1099, IEEE, 2009.

[10] B.-K. Cho, I.-W. Park, and J.-H. Oh, "Running pattern generation of humanoid biped with a fixed point and its realization," *International Journal of Humanoid Robotics*, vol. 6, no. 4, pp. 631–656, 2009.

[11] C. Chevallereau, E. Westervelt, and J. Grizzle, "Asymptotically stable running for a five-link, four-actuator, planar bipedal robot," *The International Journal of Robotics Research*, vol. 24, no. 6, pp. 431–464, 2005.

[12] I. Poulakakis and J. Grizzle, "Modeling and control of the monopedal robot Thumper," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 3327–3334, IEEE, 2009.

[13] A. D. Ames, "First steps toward underactuated human-inspired bipedal robotic walking," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 1011–1017, IEEE, 2012.

[14] K. Sreenath, H.-W. Park, I. Poulakakis, and J. Grizzle, "Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on MABEL," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 324–345, 2013.

[15] S.-H. Hyon and T. Emura, "Energy-preserving control of a passive one-legged running robot," *Advanced Robotics*, vol. 18, no. 4, pp. 357–381, 2004.

[16] M. Ahmadi and M. Buehler, "Controlled passive dynamic running experiments with the ARL-monopod II," *Robotics, IEEE Transactions on*, vol. 22, no. 5, pp. 974–986, 2006.

[17] M. Raibert, K. Blankespoor, G. Nelson, R. Playter, *et al.*, "Bigdog, the rough-terrain quadruped robot," in *Proceedings of the 17th World Congress*, pp. 10823–10825, 2008.

[18] J. K. Hodgins and M. Raibert, "Adjusting step length for rough terrain locomotion," *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 3, pp. 289–298, 1991.

[19] U. Saranli, W. Schwind, and D. Koditschek, "Towards the control of a multi-jointed, monoped runner," in *Proceedings of the IEEE international conference on robotics and automation*, pp. 2676–2682, 1998.

[20] A. Seyfarth, H. Geyer, and H. M. Herr, "Swing-leg retraction: a simple control model for stable running," *J. Exp. Biol.*, vol. 206, pp. 2547–2555, 2003.

[21] Y. Blum, J. Rummel, and A. Seyfarth, "Advanced swing leg control for stable locomotion," in *Autonome Mobile Systeme 2007*, pp. 301–307, Springer, 2007.

[22] G. S. Carver, N. J. Cowan, and J. M. Guckenheimer, "Lateral stability of the spring-mass hopper suggests a two-step control strategy for running," *Chaos*, vol. 19, no. 2, pp. 026106–026106–14, 2009.

[23] A. Wu and H. Geyer, "The 3-d spring-mass model reveals a time-based deadbeat control for highly robust running and steering in uncertain environments," *Robotics, IEEE Transactions on*, no. 99, pp. 1–11, 2013.

[24] M. Hutter, C. D. Remy, M. Hopflinger, and R. Siegwart, "Slip running with an articulated robotic leg," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 4934–4939, IEEE, 2010.

[25] P. M. Wensing and D. E. Orin, "High-speed humanoid running through control with a 3d-slip model," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 5134–5140, IEEE, 2013.

[26] R. Blickhan, "The spring-mass model for running and hopping," *J. of Biomech.*, vol. 22, pp. 1217–1227, 1989.

[27] M. Dickinson, C. Farley, R. Full, M. Koehl, R. Kram, and S. Lehman, "How animals move: an integrative view," *Science*, vol. 288, no. 5463, pp. 100–106, 2000.

[28] J. A. Grimes and J. W. Hurst, "The design of ATRIAS 1.0 a unique monoped, hopping robot," in *Proceedings of the 2012 International Conference on Climbing and walking Robots and the Support Technologies for Mobile Machines*, pp. 548–554.

[29] H. Geyer and H. Herr, "A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 18, no. 3, pp. 263–273, 2010.

[30] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*. Springer Verlag, 2000.