Robust Spring Mass Model Running for a Physical Bipedal Robot

William C. Martin[1], Albert Wu[1], and Hartmut Geyer[1]

Abstract— **The analysis of the conceptual spring mass model for running reveals swing-leg placement policies that generate very robust locomotion in unobserved terrain with large changes in ground height. However, while this theoretical result suggests a potential for large improvements on the robustness of running machines, it has so far not been demonstrated on a physical robot. Here we address this implementation and verification step for a human-sized bipedal robot platform confined to a boom. We detail challenges and solutions for the implementation of the control approach and show that it leads to very robust running (ground changes ±20% of leg length) over unobserved ground in a high fidelity simulation of the robot platform. We also present initial tests on the actual robot hardware, which indicate the feasibility of the approach. If it can be generalized to 3D running, it could trigger running machines with largely improved robustness.**

I. INTRODUCTION

The spring mass model (SMM) is a common starting point for developing legged running controls. This conceptual model produces center of mass (COM) dynamics that closely match the whole body dynamics found in human and animal running experiments [1], [2]. Many legged robot platforms were inspired by this fact and either embed or emulate spring-like behavior in stance [3], [4], [5], [6]. Several control strategies have been combined with this stance behavior in mind, ranging from the regulation of fundamental locomotion goals (trunk orientation, total energy, and forward speed) with intuitive feedback laws [3], [7] to formal controller development of a spring-legged limit cycle gait within the hybrid zero dynamics framework [8].

Although these control strategies have led to successful locomotion in robots, the theory of the spring mass model suggests an alternative control strategy that should greatly improve the robustness of running robots in rough terrain. The analysis of the spring mass model's full behavior within the framework of optimal control has led to swing leg placement policies that render the model deadbeat stable [9], [10], [11], [12], [13], [14]. In particular, some of the control strategies can be embedded in a time-based swing leg policy, which leads to very robust running in unobserved terrain with frequent and large ground height changes [9], [14]. However, this theoretical result on the spring mass model has so far not been demonstrated on physical running robots.

Our goal is to implement the time-based swing-leg policy on a bipedal robot and to transfer and verify the running robustness predicted by the theory. To this end we have developed in previous work a control mapping approach that embeds the spring mass behavior in articulated bipeds with more degrees of freedom than the abstract model [15]. Here we address the hardware implementation of this biped control on a human-sized bipedal robot platform. We first review the general control approach from the spring mass model's swing leg policies to the mapping control and its additional requirements for articulated robots (section II). We then detail challenges and solutions for the implementation of the control approach on CMU's ATRIAS biped platform (section III), and show that in a high fidelity simulation of the robot platform the control approach leads to highly robust running capable of terrain height changes ±20% leg length (section IV). Section IV also presents our initial tests on the actual robot hardware, which indicate that the simulation performance will translate to the physical robot once the system gets upgraded with larger power amplifiers. Finally, we discuss this limitation and future plans to generalize the approach to 3D running (section V).

II. REVIEW OF CONTROL APPROACH

The spring mass model describes essential mechanics of legged running. The model includes a point mass m attached to a massless spring with stiffness k_0 and rest length l_0. During flight, the point mass moves on a purely ballistic trajectory. Once the spring leg hits the ground, the model enters stance assuming perfect ground contact. In stance, the model behaves like an inverted pendulum with an embedded spring. The resulting ground reaction forces (GRFs) are

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = k_0(l_0 - l)\begin{bmatrix} -\cos(\alpha) \\ \sin(\alpha) \end{bmatrix}, \tag{1}$$

where α is the angle the spring leg forms with the ground, and l is the current leg length. The model returns to flight when the leg is fully extended during rebound. The analysis of the model's apex return map, the stride map between apex events of two consecutive flight phases, leads to swing leg policies $\alpha^*(t)$ that commence after the apex event and generate robust running of the SMM over rough terrain without prior knowledge of the upcoming ground height [14]. The swing leg policies are unique and depend on the model's spring stiffness k_0, system energy E_{sys}, and the desired height y_{des} of the COM at apex.

To translate the robust behavior to bipedal robots, we have developed a control mapping approach in previous work that takes the additional degrees of freedom of physical robots into account [15]. The main idea of this control mapping is to produce spring-mass dynamics with the robot in stance

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

[1]All authors are with the Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA. {wmartin, apwu,hgeyer}@cmu.edu

and to follow a given policy $\alpha^*(t)$ with the leg that is about to land in swing. Several other control goals are handled in parallel. These additional goals include the regulation of a robot's system energy and trunk orientation, and the behavior of the swing leg that is *not* about to land.

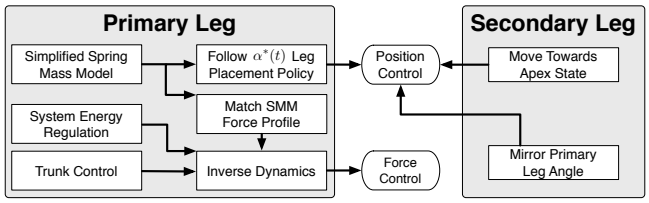

Fig. 1. The running controller implements different high-level control goals for the primary and secondary legs. These objectives change depending on whether the primary leg is in stance or swing.

Figure 1 sketches the relationship of the control goals. The goals are divided between a primary leg and a secondary leg. The primary leg is the leg whose control tracks the behavior of the spring mass model. In swing, this leg's control tries to reach the initial configuration $\alpha^*(0)$ before apex has been reached in flight and, after this event, to track the policy $\alpha^*(t)$. In stance, the control attempts to generate GRFs that match the simplified SMM, computed from (1) assuming a virtual spring between the foot point and the COM of the robot. However, energy and trunk regulation alter this GRF profile. The stiffness of the virtual leg spring is modulated throughout stance,

$$k = k_0 + c\Delta E\,\dot{l}, \tag{2}$$

based on an energy error ΔE, an energy correction gain c, and the leg lengthening velocity $\dot{l}$. This allows the control to add or remove energy from the system. In addition, the desired GRF vector is rotated by an angle θ_{rot} about the foot,

$$[F_x^*\ F_y^*]^\intercal = R_{\theta_{rot}}\,[F_x\ F_y]^T, \tag{3}$$

where $R_{\theta_{rot}}$ is the rotation matrix. This generates corrective torques about the COM and stabilizes the trunk at a desired angle θ^*. The resulting GRF profile is mapped into joint torques of the robot using inverse dynamics,

$$\mathbf{z} = \begin{bmatrix} \mathbf{M}_{5\times5} & -\mathbf{S}_{5\times2} \\ \mathbf{J}_{2\times5} & \mathbf{0}_{2\times2} \end{bmatrix}^{-1} \left(\begin{bmatrix} \mathbf{h}_{5\times1} \\ -(\dot{\mathbf{J}}\dot{\mathbf{q}})_{2\times1} \end{bmatrix} + \begin{bmatrix} \mathbf{J}^T_{5\times2} \\ \mathbf{0}_{2\times2} \end{bmatrix} \begin{bmatrix} F_x^* \\ F_y^* \end{bmatrix} \right), \tag{4}$$

where $\mathbf{z} = [\ddot{x}\ \ddot{y}\ \ddot{l}\ \ddot{\gamma}_l\ \ddot{\theta}\ \tau_f\ \tau_b]^T$ specifies commanded joint torques and accelerations (compare Fig. 3), $\mathbf{q}$ is the coordinate vector, $\mathbf{M}$ is the mass matrix, $\mathbf{J}$ is the Jacobian mapping GRFs to joint torques, $\mathbf{S}$ is the selection matrix, and $\mathbf{h}$ accounts for Coriolis, centrifugal, and gravitational forces.

The secondary leg control is only active in swing. During the swing of the primary leg (flight phase as both legs are in swing), the secondary leg mirrors the primary leg's angular position and velocity to diminish trunk pitching. When the primary leg enters stance, the secondary leg tracks the initial apex configuration $\alpha^*(0)$, preparing for a switch of roles of the two legs. The switch occurs when the primary leg leaves stance.

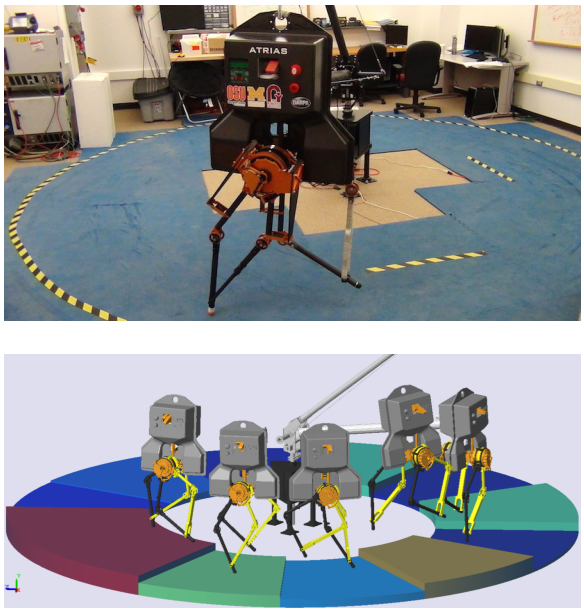

Fig. 2. Top: Carnegie Mellon's ATRIAS robot testbed for planar running. Bottom: Simulation of the ATRIAS testbed used to pre-tune control and provide performance benchmarks for the hardware implementations.

III. IMPLEMENTATION ON ATRIAS

The realization of the control goals on specific robot hardware poses challenges that revolve around joint torque distribution, force and position control of desired behaviors, contact detection, and estimation of system energy. We detail how we address these challenges for the implementation on CMU's ATRIAS biped (Fig. 2).

A. Robot System

ATRIAS is a human-size bipedal robot developed by the Dynamic Robotics Laboratory at Oregon State University [16]. The robot's legs are composed of lightweight four-bar mechanisms. The proximal segments of each four-bar mechanism are actuated through series fiberglass springs by geared electric DC motors. The torques generated by these series elastic actuators (SEAs) effectively produce leg length l and leg angle θ_l actuation. Each leg is further actuated by a geared DC motor housed in the trunk that provides leg abduction and adduction. To transfer the planar control of the spring mass model, the trunk is constrained by a boom, although it remains free to pitch in the sagittal plane. The lightweight leg construction with four-bar mechanisms concentrates the robot's mass distribution. It approximates a spring mass model with a leg length of about 1m and a mass of about 60kg. Key differences to the spring mass model are the robot's COM location about 0.1m above the hips, the trunk's rotational inertia of about 2.2 kg·m^2, and the four-bar motors' reflected inertia of about 3.75 kg·m^2 when the robot is in stance.

We have developed a detailed simulator of the ATRIAS testbed in Matlab SimMechanics (Mathworks, Natick, MA) for pre-tuning the control and providing performance benchmarks for the hardware implementations (Fig. 2). The simulated environment contains 13 degrees of freedom (4 for

the four-bar mechanisms, 4 for the SEA motor positions, 2 for the frontal plane motors, 1 for the trunk pitch, and 2 for the boom constraining roll and yaw) and models the main mechanical components (segmented chain, gear stages, springs, motor dynamics) and electrical components (electrical motor dynamics, discrete time control) of the robot testbed. In addition, the simulation includes contact points on the robot's feet, modeling the dissipative ground reaction dynamics as nonlinear spring-dampers with stick-slip transitions [17]. Gaussian sensor noise and joint friction are also included to create a realistic system.

B. Torque Limitations

Each four-bar motor of ATRIAS develops about 300Nm peak torque at joint level (gear ratio 50:1). Running on rough terrain forces the robot to operate at motor torque limits, leading to implementation challenges with torque saturation.

1) Optimal Force-Torque Distributions: Torque saturation occurs mainly in stance when trading off leg force generation and trunk pitch stabilization. We use optimal control to find the best compromise. A combination of desired leg force, F_l, and hip torque, τ_H, translates into desired torques for the front and back segments of the four-bar leg,

$$\tau_{f/b} = \tau_H/2 \pm F_l l_s \sin(\rho/2),$$

where l_s is the four-bar segment length and ρ is the splay angle (Fig. 3). The optimal torque distribution problem is formulated with objectives $x_1 = \tau_H/2$ and $x_2 = F_l l_s \sin(\rho/2)$ and the cost function $f(x_1, x_2) = -\lambda_1(x_1 - x_1^*)^2 - \lambda_2(x_2 - x_2^*)^2$, where the positive parameters λ_1 and λ_2 shift the control emphasis between leg force and trunk pitch. The pair x_1^* and x_2^* describes objective values computed from the desired leg force and hip torque. Combining the resulting first-order and complementary slackness conditions produces a system of equations with nine possible solutions. Based on $x_1^*, x_2^*, \lambda_1, \lambda_2$ and the specified torque limits, we select the solution which maximizes the objective function.

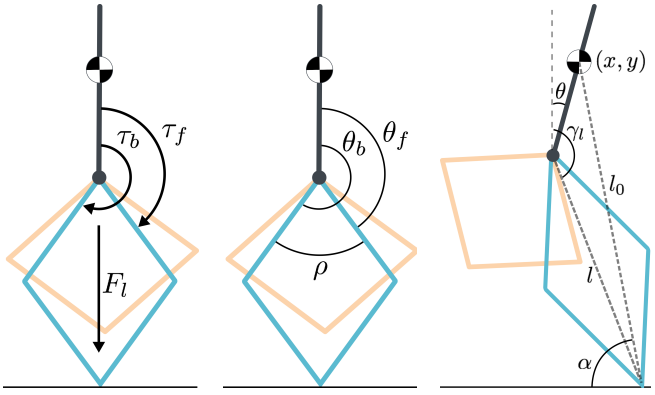

Fig. 3. Inverse dynamics control during stance computes front and back torques τ_f and τ_b that produce leg force F_l and accelerations for COM position $(\ddot{x}, \ddot{y})$, leg length $\ddot{l}$, primary leg angle $\ddot{\gamma}_l$, and torso acceleration $\ddot{\theta}$. Leg placement during the flight phase controls virtual spring angle α with rest length l_0.

2) Feasible SMM Control: Motor saturations also limit the peak values of leg force and hip torque that can be achieved. The maximum leg force, F_{max}, decreases in proportion to the four-bar splay ρ. The splay can be limited during rebound in stance by commanding a large virtual spring stiffness k_0 that reduces leg compression. However, a large spring stiffness reduces the apex height or ground clearance that the robot can achieve. This relationship between virtual spring stiffness, ground clearance, and peak motor torque is shown in Fig. 4 for vertical hopping of the robot. The contour line at 80% peak motor torque highlights stiffness-clearance combinations that the robot can safely achieve. For implementation on the robot, we select the virtual stiffness (k_0 = 8820N·m^{-1}) that maximizes ground clearance (4.7cm), in order to avoid motor saturations when encountering ground height changes in rough terrain locomotion.

The maximum hip torque limits how well the robot can track swing-leg angle policies $\alpha^*(t)$ provided by the spring mass model. The leg angle policies require large hip speeds when system energy E_{sys} decreases. Thus, we select a system energy $E_{sys} \geq 445J$ that is large enough to produce leg angle policies within the tracking capabilities of the robot.

In consequence, we target the behavior of a spring mass system with mass m = 60kg, leg rest length l_0 = 1.04m, leg stiffness k_0 = 8820N·m^{-1}, system energy E_{sys} = 445J, and target apex height y_{des} = 1.06m. With these parameters, the swing leg policy for robust running on unobserved rough terrain can be approximated to $\alpha^*(t) = 12.835t^3 - 2.9736t^2 + 1.9343t + 1.1389$ [9].

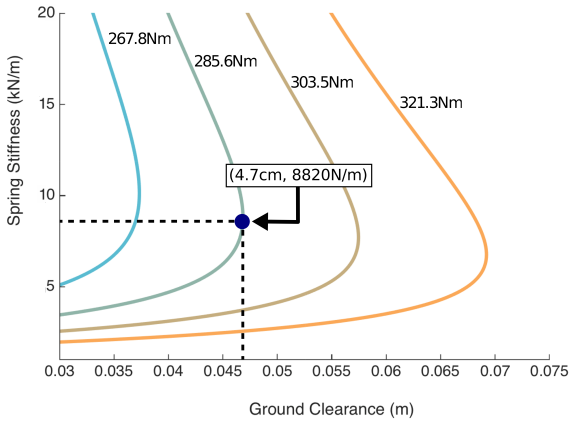

Fig. 4. Contour lines for stiffness-clearance combinations that can be achieved at 75%, 80%, 85%, and 90% of peak motor torque.

C. Torque and Position Tracking

ATRIAS' four-bar motors act through carbon fiber leaf springs on the proximal leg segments. The switch between stance and swing phases produces very different load environments, with high load inertia in stance and very low load inertia in swing, leading to implementation challenges with series elastic actuator control. We address these challenges by gain-scheduling and switching between torque and position control based on the experienced load.

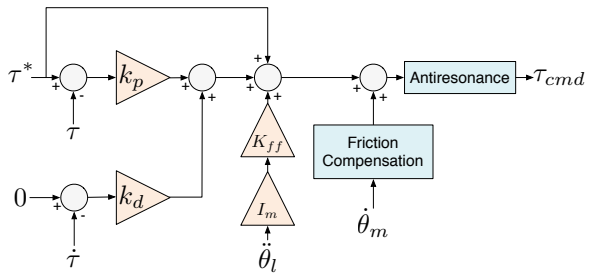

Fig. 5. Torque control for series elastic actuators includes friction compensation and antiresonance terms. Feedback gains, k_p and k_d, are scheduled based on measured torque. Load acceleration $\ddot{\theta}_l$ along with motor inertia I_m provide a feedforward term that is scaled by the constant K_{ff}.

1) Torque Tracking: The last two rows of equation (4) are solved to provide desired joint torques τ^* for the SEAs. To track these torques in stance, the SEA control uses classic design [18] with modifications (Fig. 5). The desired torque is tracked combining PD feedback on torque error $e = \tau^* - \tau_{mea}$ with feedforward compensation of the SEA dynamics. The measured torque $\tau_{mea} = k_{sea}(\theta_m - \theta_l)$ is obtained from motor and joint position encoders and the leaf spring stiffness k_{sea}. The feedback gains are scheduled based on this measured torque,

$$k_p = \frac{k_p^{\max}}{1 + \exp(\tau_{th} - |\tau_{mea}|)},\ k_d = \frac{k_d^{\max}}{1 + \exp(\tau_{th} - |\tau_{mea}|)}, \quad (5)$$

where τ_{th}=80N·m is a torque threshold for switching from low to high gains, and $k_p^{\max}$ and $k_d^{\max}$ are maximum gains. This gain scheduling prevents large spring vibrations for small loads. In addition, Coulomb friction torque is compensated assuming $\tau_{cf} = a + b\tanh(k\,\dot{\theta}_m)$ with identified friction constants a and b, a sensitivity gain k, and the motor velocity $\dot{\theta}_m$. Viscous friction is negligible and ignored. Finally, the commanded motor torque is passed through a notch filter at 27Hz to further prevent resonance in the leaf spring. ATRIAS' motor controllers convert the commanded motor torque into appropriate current at the lowest control level.

2) Position Tracking: The very low load inertia provided by ATRIAS' carbon fiber four-bar mechanism (total weight about 1kg) in combination with random, unmodeled friction torques of the actuators make it difficult to implement torque tracking in swing. Thus, we use position tracking for swing leg control goals. For both, the primary and secondary leg, these goals require to track desired hip angle $\phi_h^*(t)$ and leg length $l^*(t)$. The corresponding joint targets for the front and back segment of the four-bar mechanism are given by

$$[\theta_f^*\ \theta_b^*]^T = \phi_h^* \pm \arccos\left(\frac{l^*}{2l_s}\right). \quad (6)$$

The joint targets are tracked with PD control and the feedforward compensation term $I_r\ddot{\theta}_{f/b}^*$, where I_r is the reflected motor inertia (Fig. 6). In addition, friction compensation and notch filtering are preserved from the SEA torque control.

The desired hip angle and leg length are simplest for the secondary leg mirroring the primary leg. In this case, $l^*(t)$ = 0.7m is set to a constant retraction value and $\phi_h^{*s}(t) = 2\pi -$

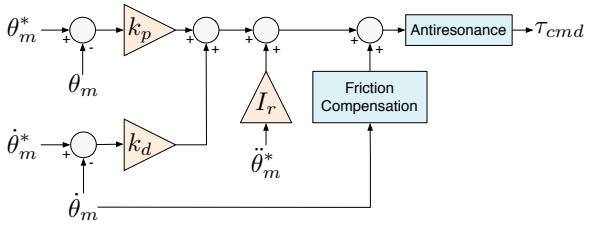

Fig. 6. Position control during leg swing is performed using a PD law along with feedforward compensation for desired accelerations. Friction and resonance compensations are also included.

$\phi_h^{*p}(t)$ mirrors the hip angle of the primary leg about the robot's symmetry axis (Fig. 7A).

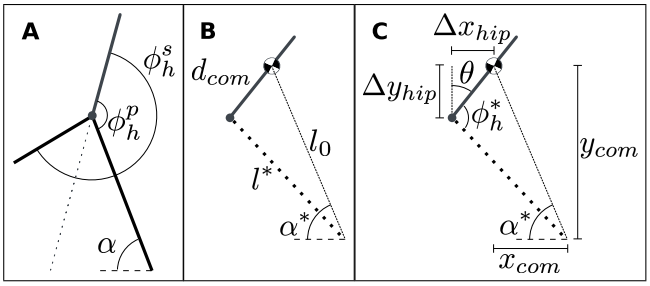

Fig. 7. **A**: The secondary leg mirrors the primary leg during robot flight by reflecting primary leg angle ϕ_h^p across the trunk centerline. **B**: Leg length l^* can be calculated for a given α^* using the law of cosines. **C**: Relative leg angle ϕ_h^* can be resolved from a given α^* by solving for the hip point coordinates $(x_{com} + \Delta x_{hip},\ y_{com} - \Delta y_{hip})$.

In the case of the primary leg following the swing leg policy $\alpha^*(t)$ after apex, the desired hip angle and leg length resolve to

$$\phi_h^* = (\pi/2 - \theta) + \text{atan}(\Delta y/\Delta x), \quad (7)$$

$$l^* = \sqrt{d_{com}^2 + l_0^2 - 2d_{com}l_0\cos(\theta - \alpha^* + \pi/2)}, \quad (8)$$

where d_{com} is the distance between COM and hip, $\Delta x = l_0\cos\alpha^* + d_{com}\sin\theta$, and $\Delta y = l_0\sin\alpha^* - d_{com}\cos\theta$ (Fig. 7B, 7C).

Finally, if either the primary or the secondary leg try to reach the initial configuration $\alpha^*(0)$, the desired trajectories of the leg length and hip angle are generated as cubic polynomials $l^*(t) = a_l t^3 + b_l t^2 + c_l t + d_l$ and $\phi^*(t) = a_\phi t^3 + b_\phi t^2 + c_\phi t + d_\phi$, where $t \in [0\ t_a]$ and t_a is the anticipated time until the apex event. The coefficients of the polynomials are solved for to match the initial and final positions and velocities, with the final values generated from (7) and (8) using $\alpha^* = \alpha^*(0)$. The resulting trajectories diminish discontinuities in the desired velocities and improve the tracking accuracy of the swing target policy $\alpha^*(t)$ after the apex event.

D. Event Detection and Parameter Estimation

The final implementation challenges exist for the contact detection and the estimation of the system energy and the apex event time.

1) Ground Contact Detection: In the current version, the ATRIAS robot does not have dedicated contact switches on the feet. To detect contact and switch between stance and

swing control, we rely on a low-pass filtered estimate of the vertical ground reaction force, $GRF_y = F_l \cos(\phi_h + \theta)$, with the leg force measured from the spring deflections of the SEAs, $F_l = (\tau^f_{mea} - \tau^b_{mea})/(2l_s \sin(\rho/2))$.

Based on this estimate, reliable switches between stance and swing are ensured with a state machine detecting key events (Fig. 8). When GRF_y exceeds a touch-down threshold F_{th_1} (20% body weight), the leg switches from swing to stance control. To prevent foot scuffing from triggering false contact events, the leg remains locked in stance control only if a second threshold F_{th_2} (50% body weight) is passed. To exit this stance, the leg has to compress and rebound, during which a drop of GRF_y below the 50% threshold triggers take-off.

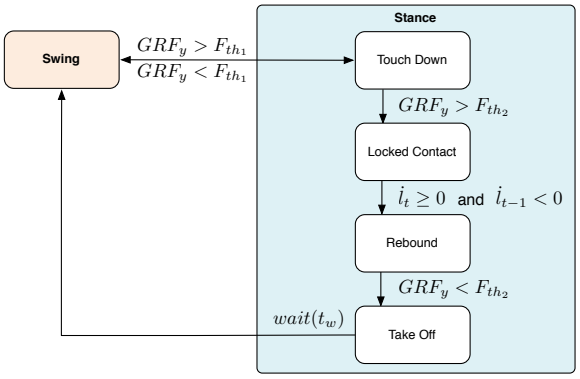

Fig. 8. Reliable switches between stance and swing control states.

2) System Energy: A swing leg policy $\alpha^*(t)$ for robust locomotion is valid for a particular system energy E_{sys} of the spring mass model in flight. This energy is tracked for ATRIAS in stance by adapting the virtual leg stiffness (2) based on the energy error $\Delta E = E_{sys} - E$. E is the equivalent spring-mass energy $E = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) + mgy$ of the robot in the preceding flight phase, where m is the total robot mass, g is the gravitational acceleration, $\dot{x}$ and $\dot{y}$ are the horizontal and vertical velocity of the center of mass at take off, and y is the take off height. The values of $\dot{x}$, $\dot{y}$ and y are estimated from the robot kinematics at the take off event in stance (Fig. 8) using 100ms moving average filters.

3) Time of Apex: The swing leg policy $\alpha^*(t)$ is triggered at apex. Like system energy, we estimate the time of the apex using the take off event. From this event, the time it takes to reach apex is estimated as $\Delta t = \dot{y}/g$.

IV. RESULTS AND DISCUSSION

The implementation of robust running control based on the theory of the spring mass model introduced a third control layer to deal with the specific ATRIAS robot platform and its limitations (1st: spring mass model [9], [14]; 2nd: control mapping to articulated biped [15]; both reviewed in section II). To provide a benchmark for what this physical platform should be capable of with this layered control, we first present results on level and rough terrain running for the high fidelity ATRIAS simulation, and then discuss our preliminary results on the actual hardware.

A. Simulation Benchmark Results

The current motor torque saturation of 300N·m limits ATRIAS's performance. Over flat ground, the simulated robot can maintain a constant target apex height with a mean and standard deviation error of 1.061 ± 0.005m at a running speed of 2.091 ± 0.017m·s^{-1} (Fig. 9). The same controller tolerates small terrain height variations up to ±4% of leg length (not shown in Fig. 9).

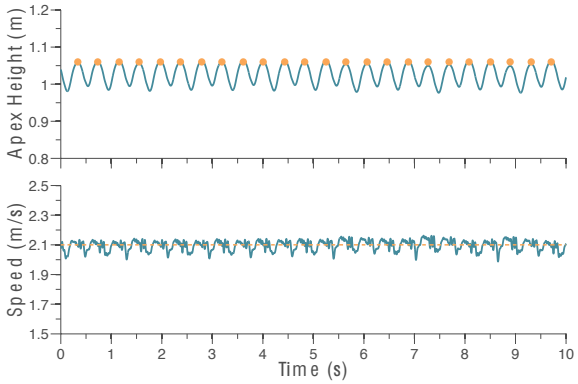

Fig. 9. The simulated system demonstrates steady state running over level ground with constant apex height and forward velocity.

The ATRIAS platform will receive upgrades to its motor controllers, however. This upgrade will increase the peak motor torques to about 700N·m. Figure 10 shows the tracking performance that the layered control should achieve with this upgrade. The simulated robot is capable of robust running over rough terrain up to 20% of maximum leg length at a running speed of about 2.467 ± 0.145m·s^{-1}. The variation in height and speed occur due to system energy corrections applied during stance. Note that the COM apex height relative to the ground height at take off is maintained at 1.146 ± 0.102m. This performance is within 80% of the performance observed for the theoretical spring mass model and for the control mapping applied to an ideal biped model [15].

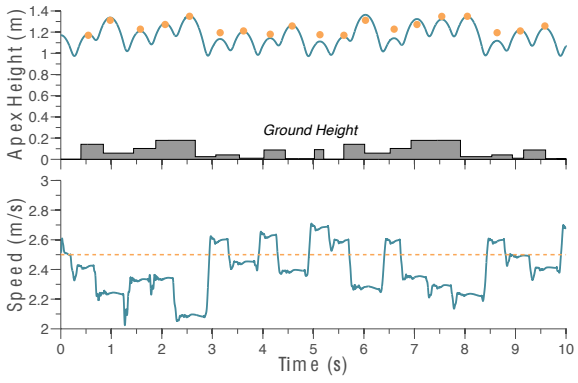

Fig. 10. Increased motor torque saturations allow the simulated robot to handle rough terrain approximately 20% of leg length. System energy errors are corrected between strides as the ground rises and lowers, adjusting the target apex heights.

B. ATRIAS Hardware Experiments

The transfer of the controller to the robot hardware is an ongoing process, which we approach in incremental steps. At the current stage, the embedded control includes the virtual spring leg behavior in stance with trunk and energy regulation, swing leg mirroring, and the contact detection (section III). The swing leg placement policy $\alpha(t)^*$ is not yet implemented. Instead, we use a basic leg placement strategy for evaluation purposes.

With this preliminary work, the robot shows virtual spring hopping behavior over flat terrain with a consistent apex height of 1.091±0.009m at a forward speed of 0.444±0.088m·s^{-1} (Fig 11). The achieved hopping performance is close to what is observed in simulation when using the same control implementation stage. Despite the current torque limitations, the real system is capable of significant ground clearance close to 8cm.

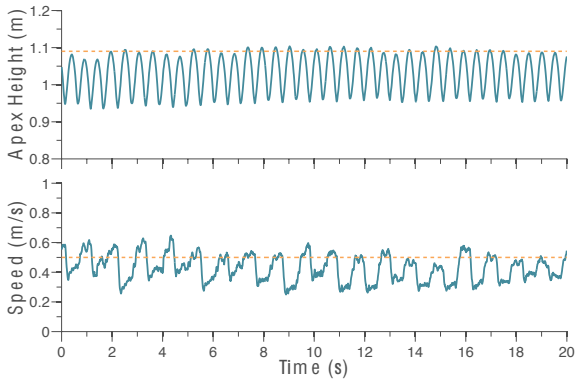

Fig. 11. The real robot hardware is currently capable of spring-like hopping over flat terrain with consistent apex heights.

V. CONCLUSIONS AND FUTURE WORK

Our goal is to demonstrate and verify with a human-sized bipedal robot highly robust running based on the swing-leg placement policies derived from the theoretical spring mass model. In this paper, we addressed challenges and solutions for this implementation step of the control approach and showed that it leads to very robust running over unobserved ground in a high fidelity simulation of the robot platform. In addition, initial tests on the actual robot hardware indicate the feasibility of the approach. However, more implementation work is needed to confirm or refute the theoretical results in robot experiments.

This work is currently ongoing. We are implementing on the robot hardware the swing leg control policy for running at about 2.0m·s^{-1}, which in simulation was achievable within the current motor performance limits of ATRIAS. In addition, the robot platform is getting upgraded, which will more than double the motor torque outputs and allow us to test the running robustness of the robot on large ground variations. Ultimately, we seek to generalize the control mapping [15] and implementation to 3D, and to demonstrate highly robust running over unobserved terrain outside of the laboratory environment.

ACKNOWLEDGMENT

We thank Jessy Grizzle and his group at the University of Michigan for helpful conversations on the optimal torque distribution of the four-bar motors.

REFERENCES

[1] R. Alexander, "Three uses for springs in legged locomotion," *Int. J. Rob. Res.*, vol. 9, pp. 53–61, 1990.

[2] M. H. Dickinson, C. T. Farley, R. J. Full, M. A. R. Koehl, R. Kram, and S. Lehman, "How animals move: an integrative view," *Science (80-.).*, vol. 288, no. 5463, pp. 100–106, 2000.

[3] M. H. Raibert, *Legged robots that balance*. MIT press, Cambridge, 1986.

[4] G. Zeglin and B. Brown, "Control of a bow leg hopping robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1998, pp. 793–798.

[5] J. W. Grizzle, J. Hurst, B. Morris, H.-W. Park, and K. Sreenath, "MABEL, a new robotic bipedal walker and runner," in *Am. Control Conf. 2009. ACC'09.* IEEE, 2009, pp. 2030–2036.

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

[7] M. Raibert, K. Blankenspoor, G. Nelson, and R. Playter, "BigDog, the Rough-Terrain Quadruped Robot," in *Proc. 17th World Congr. Int Fed Autom. Control*, 2008, pp. 10 823–10 825.

[8] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, "Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on MABEL," *Int. J. Rob. Res.*, vol. 32, no. 3, pp. 324–345, 2013. [Online]. Available: http://ijr.sagepub.com/content/32/3/324.abstract

[9] 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.

[10] 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. 26 106–26 114, 2009.

[11] O. Arslan, U. Saranli, and O. Morgul, "Reactive footstep planning for a planar spring mass hopper," in *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, 2009, pp. 160–166.

[12] 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*. IEEE, 2010, pp. 4934–4939.

[13] B. Andrews, B. Miller, J. Schmitt, and J. E. Clark, "Running over unknown rough terrain with a one-legged planar robot," *Bioinspir. Biomim.*, vol. 6, no. 2, 2011.

[14] 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," 2013.

[15] ——, "Highly robust running of articulated bipeds in unobserved terrain," in *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*. IEEE, 2014.

[16] 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*, 2012, pp. 548–554.

[17] H. Geyer and H. Herr, "A Muscle-Reflex Model that Encodes Principles of Legged Mechanics Produces Human Walking Dynamics and Muscle Activities," *IEEE Trans Neural Syst Rehabil Eng*, vol. 18, no. 3, pp. 263–273, 2010.

[18] G. A. Pratt and M. M. Williamson, "Series elastic actuators," in *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 1. IEEE, 1995, pp. 399–406.