An Analytically Computable Model and Energy-Based Control for a Realistic Series-Elastic 1D Hopper on Rough Terrain

Pat Terry and Katie Byl

***Abstract*— In this paper, we describe modeling and control techniques to accurately representing a real-world series-elastic actuated 1D hopping robot. There is an abundance of work regarding the implementation of highly simplified hopper models, with the hopes of extracting fundamental control ideas for running and hopping robots. Accurately controlling a non-linear system such as the hopper depends on a good forward model, and the inability to analytically solve even an idealized 2D case often leads to a largely simplified analysis, such as the classic SLIP model. However, real-world systems unfortunately cannot be accurately described by such simple models, as real actuators have their own dynamics including additional inertia and non-linear frictional losses. Therefore, an important step towards demonstrating high controllability and robustness to real-world, uneven terrain is in providing accurate higher-order models of real-world hopper dynamics. Motivated by actual hardware, our work here addresses 1D modeling and verification of the real-world dynamics of a hopper designed for energy efficiency, with the eventual goal of developing robust and agile control for 2D and 3D hoppers utilizing accurate feed-forward predictions of dynamics.**

I. Introduction

Hoppers are useful models for studying methods by which legged robots can navigate and select footholds on intermittent, rough terrain, and they also have obvious potential for both fast and energy efficient locomotion. Bridging the gap between highly idealized spring-loaded inverted pendulum (SLIP) models and the higher-order, lossy, real-world dynamics of hopping robots remains an open question. This work aims to facilitate application of approximate, closed-form planning and control solutions for hopper models to real-world robots by providing an exact, closed-form solution for a real-world, 1D hopper system.

The potential benefits of legged locomotion are many-fold and have been an area of active study for many years. Using compliant springs as energy storage devices for legged locomotion is a concept that has been extensively studied [1], and has lead to the application of the spring-loaded inverted pendulum (SLIP) model, which is a simplistic model that is used to help describe the dynamics of many legged animals and legged robots, and most particularly, of hopping robots. Much work has been done in the area of legged locomotion utilizing SLIP [2] [3] due to its modeling simplicity. Studying the behavior of a single leg and extending this to multiple legs has been shown to be effective [4] and has lead to the development of many hopping and other leg-actuated robots, including well-known work led by Raibert [5]. One of the main difficulties in modeling a 2D hopper with SLIP is the inability to find a complete analytical solution. Analytical solutions are particularly desirable to have in real-time control due to their ease of implementation and fast processing times. Approximating SLIP solutions has been an active area of study [6] [7] [8], and many of these methods focus on simplification of system models that have dynamics importantly different from those of a real hopping robot. Operation of actual hardware ultimately requires leg placement on irregularly spaced footholds on potentially rough terrain [9], and for this to be achieved, accurate state tracking is desirable. An important step in achieving this is regulation of total energy (i.e., hopping height) on rough terrain, most particularly when the ground is modeled as an unknown height disturbance on a per-step basis.

We have two primary goals in creating and validating an analytic model for a real-world hopper. First, our group is developing improved closed-form approximations for the step-to-step dynamics of planar hopping [12], [11], toward computationally fast planning with a limited, noisy look-ahead on rough terrain. We note that achieving a closed-form solution for the full dynamics of a 2D hopper is not likely, as this does not yet exist even for more idealized hopper models; however, we anticipate our analytic 1D solution will improve our closed-form step-to-step approximations of real-world hopper dynamics significantly. Second, accurate planning for 2D and 3D hoppers requires good feedforward control; since contact with the ground is brief (e.g., less than 0.25 sec), control should not rely on feedback, alone. Our system ID and modeling provide a forward model for generating such feedforward commands.

We have developed two vertically constrained hoppers [10] and are in the process of developing a 2D hopper on a boom. Fig. 1 shows the two vertical hoppers, Hopper B and Hopper C, along with the schematic used for modeling their dynamics. Each robot has an actuator in series with a spring to allow the addition of energy in the stance phase. Leg actuation is achieved via a ball screw driving a metal plate for Hopper B and by a cable-pulley network for Hopper C. An overly-simplified model is not sufficient to accurately represent real system behavior, as real actuators have significant mass and inertia and cannot respond instantaneously to set desired leg lengths. In our modeling, the state x_1 represents the actuated displacement of one end of the spring by the current-driven actuator with effective mass m_{eff}, and the second state, x_2, represents the global height of the hopper's body with sprung mass, m_s.

This work is supported by DARPA grant no. W911NF-11-1-0077.

P. Terry and K. Byl are with the Robotics Lab in the ECE Dept, University of California, Santa Barbara, CA 93106 USA pwterry at umail.ucsb.edu, katiebyl at ece.ucsb.edu

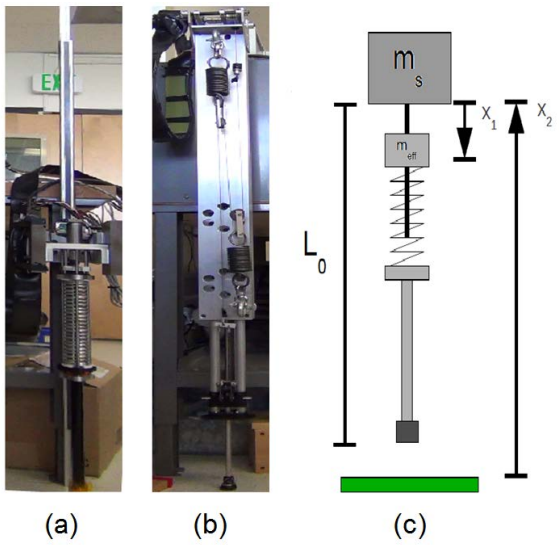

Fig. 1. (a) Hopper B, (b) Hopper C, and (c) schematic for hopper model.

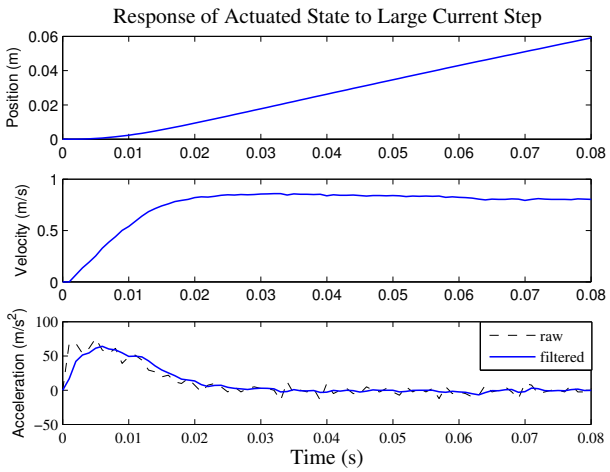

Fig. 2. Step response of the Hopper C hardware.

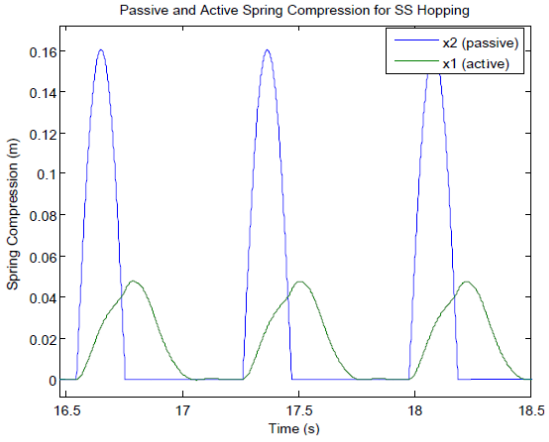

Fig. 3. Compression cycles during stance phase for steady-state operation of the Hopper C hardware.

Inclusion of effective mass and friction of the actuator is important when modeling a realistic hopper. Consider Fig. 2, which shows Hopper C's response for the largest current step input deliverable by our system for a brief time. This illustrates the physical limits of our real hopper, where it is clear the maximum velocity at which the actuator can displace x_1 is limited. Due to these limitations, the assumption that length of the leg can be set either instantaneously or with constant acceleration for desired stance phase trajectories is not valid. Compression to a desired value during stance is clearly subject to important dynamic effects, as verified on the actual hardware in Fig. 3, where in fact x_1 is still undergoing compression when the stance phase has ended.

The rest of the paper is organized as follows. In Section II, we present an early model used to develop initial control and system identification results for Hopper B and illustrate the need for a better model and feedforward control. In Section III, we propose an improved system model that takes into account additional friction, provide verification on Hopper C hardware, and derive analytic expressions for states and energy over time. Section IV shows how we use this new model to develop accurate feedforward control of apex heights, and in Section V, we summarize results using our analytic expressions for control of our hopper model.

II. Initial System Modeling

The equations of motion for the stance phase were derived using a Lagrangian approach as

$$\ddot{x}_1 = \frac{1}{m_{eff}}(k(x_2 - x_1 - L_0) - b_1\dot{x}_1 - \gamma_2 u) \qquad (1)$$

$$\ddot{x}_2 = \frac{1}{m_s}(k(x_1 - x_2 + L_0 + c) - b_2\dot{x}_2 - m_s g) \qquad (2)$$

Although mechanically different, later sections show the same model is general enough to accurately describe both Hopper B and Hopper C. Initial work with this model was performed exclusively on the Hopper B. System identification was performed by locally optimizing parameters in the model to minimize the error for static drop tests on x_2 and step responses on x_1. Even though this model takes into account some actuator limits, it proves insufficient to match the real system with good accuracy. This is illustrated in Fig. 4, where although the passive system matches reasonably well during hopping tests, significant error for low-amplitude hops is evident. Additionally, the actuated state does not match the model accurately for any parameters. In order to benchmark effectiveness of the model toward control, a controller was designed to regulate apex heights of steady-state hopping. The controller uses experimentally determined return maps in the compression phase, seen in Fig. 5, and a feedback controller in the expansion phase that uses experimental data to generate and track trajectories for $\dot{x}_2$.

Results of this control strategy, shown in Figures 6–8, demonstrate that this initial control and modeling is not practical for accurate height regulation in the presence of disturbances. Because the control is based completely on

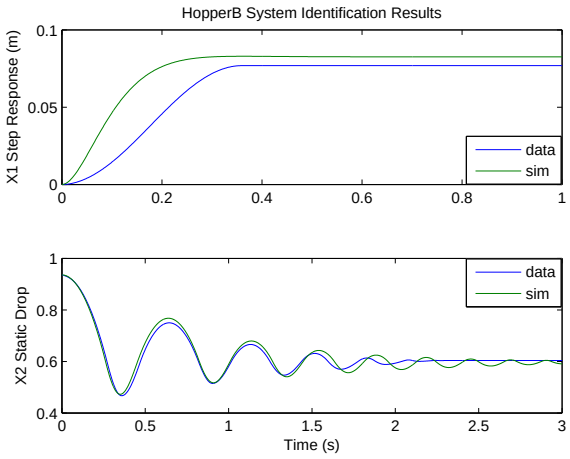

Fig. 4. System model verification for the initial Hopper B model, comparing experimental and simulated drop test data for x_2 and step response data for x_1.

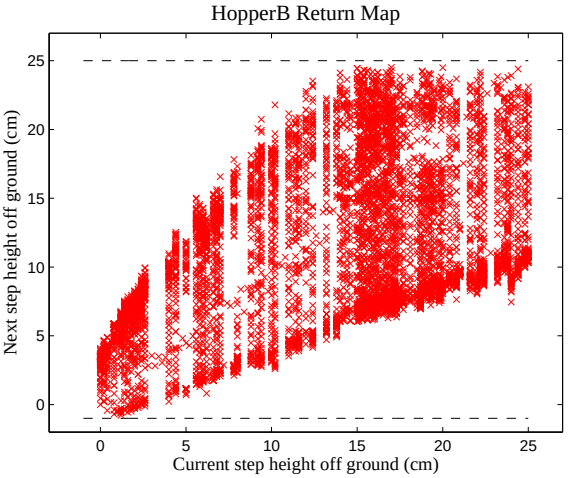

Fig. 5. Apex reachability map for Hopper B. Data represent overlays of a family of return maps, for different control actions.

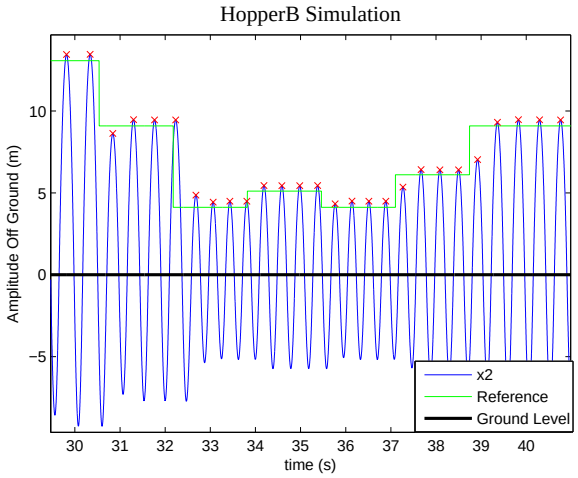

Fig. 6. Simulation results for Hopper B controller tracking references.

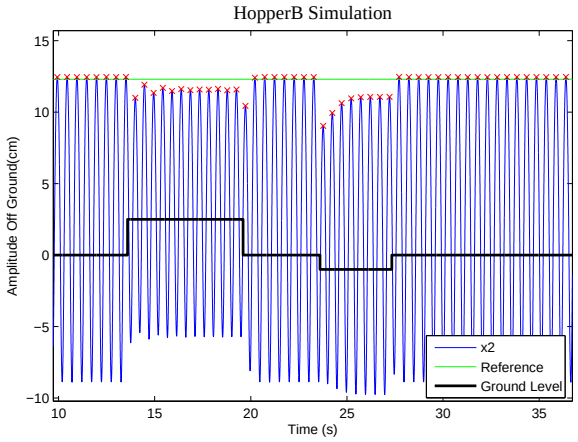

Fig. 7. These simulation results show the inability of Hopper B's initial controller to function well on uneven terrain.

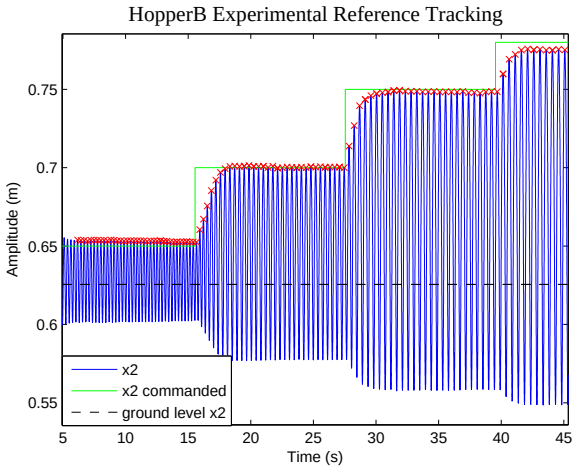

Fig. 8. Experimental results for the Hopper B controller.

experimentally mapped compression levels for even terrain only, neglecting the internal state of the actuator at touchdown, there will always be error when operating on uneven terrain. Using a purely feedback control algorithm for the physical Hopper B robot also results in a non-zero rise time for tracking references. All of these issues suggest a need for feedforward control, and the following sections of this paper demonstrate that with more complete modeling, a feedforward controller can be developed to provide precise vertical height regulation even on rough terrain. This is an important step in modeling a real system for a hopping robot.

III. Improved System Modeling

A. Equations of Motion with Added Static Friction Terms

The improved system model is motivated by previous work with Hopper B and includes both viscous and Coulombic friction terms.

$$\ddot{x}_1 = \frac{1}{m_{eff}}(k(x_2 - x_1 - L_0) - b_1\dot{x}_1 - \gamma_2 u) - f_1 sign(\dot{x}_1) \quad (3)$$

$$\ddot{x}_2 = \frac{1}{m_s}(k(x_1 - x_2 + L_0 + c) - b_2\dot{x}_2 - m_s g) - f_2 sign(\dot{x}_2) \tag{4}$$

B. Model Verification

Verification that the new model accurately represents the behavior of Hopper C was accomplished by performing system identification. Model parameters were locally optimized to minimize the error between simulated and actual static drop tests for x_2, including a 5% energy loss term at each impact, to account for unsprung foot mass, and between step responses for x_1. The static friction term f_1 can be determined by measuring the input current threshold u_* required for motion of x_1 to occur as seen in Fig. 9, and evaluating (5).

$$f_1 = u_* \frac{\gamma_2}{m_{eff}} \tag{5}$$

System identification results are shown in Fig. 10. Although some error is still present, this model yields a significant improvement in both the actuated state and, due to the additional static friction terms, the steady-state value of the passive state. The map of achievable next apex heights as a function of current apex height was determined both experimentally for the Hopper C robot and analytically using the state equations, as shown in Fig. 11. These data show the reachability of the hopper, where the bottom curve represents a drop with no actuation, and the top curve represents the approximate maximum compression possible, where x_1 is intentionally limited to within 1-2cm of bottoming out experimentally. The new model correctly captures the behavior of the bounding line, although some error is still present, likely due to measurement error and additional friction on the metal track that constrains Hopper C to 1D motion.

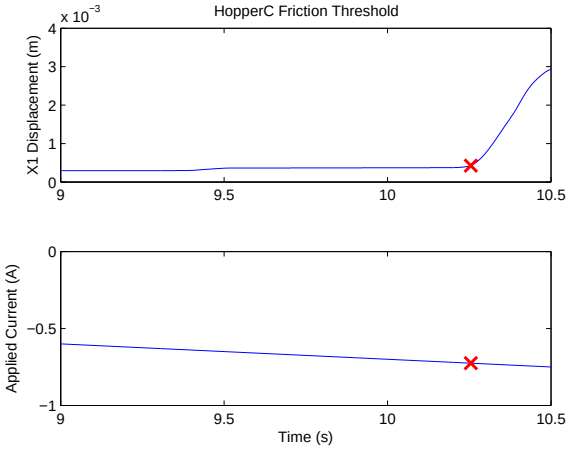

Fig. 9. Experiment used to verify and determine the static friction coefficient for the actuated state. At bottom, current is slowly increased in magnitude until motion of the actuator (at top) is detected; the required force is linearly related to current.

C. Analytical Solution of States

One of the advantages of modeling the 1D case for a vertical hopping robot is that an analytical solution exists and can be determined exactly. Thus, at touchdown a trajectory

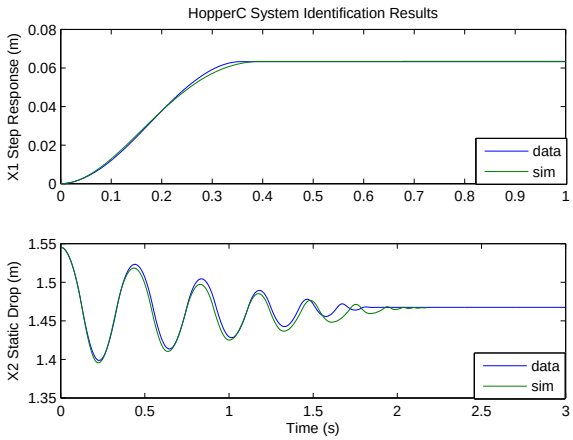

Fig. 10. Step response and drop test data, to verify system identification results of Hopper C using the improved model. Compared with Fig. 4, both transient and steady-state response for x_1 (top) are improved, and Coulombic friction effects on body height x_2 (bottom) are clearly shown.

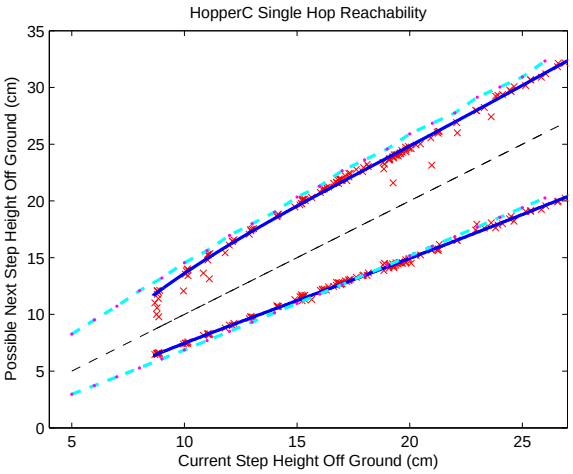

Fig. 11. Current apex to next apex reachability map for Hopper C, where the blue line and red markers represent experimental data, and the cyan dashed line and magenta markers represent analytical calculations.

can be generated based on initial conditions that characterizes the states of the system for all future time. The solution generated in this paper follows from two assumptions. Assumption A: The input current is a step change. A new solution can be generated from any point given new initial conditions and current step magnitude. Assumption B: $\dot{x}_2$ remains negative during compression, and positive during expansion. This allows the solution to be broken up into two pieces in order to overcome the non-linearity of the static friction term. If $\dot{x}_1$ does not remain positive sign during the stance phase, the equations must be broken up into N additional pieces, where N is the number of times $\dot{x}_1$ crosses the zero-axis.

The solution is generated using a Laplace Transform technique. Taking the Laplace Transform of (3) and (4) yields

$$X_1(s) = \frac{s^2 x_1(0) + s(\dot{x_1}(0) + \frac{x_1(0)b_1}{m_{eff}}) + \lambda_2 + sX_2(s)\frac{k}{m_{eff}}}{s(s^2 + s\frac{b_1}{m_{eff}} + \frac{k}{m_{eff}})} \tag{6}$$

$$X_2(s) = \frac{s^2 x_2(0) + s\beta - \gamma + s\frac{k}{m_2}X_1(s)}{s(s^2 + s\frac{b_2}{m_s} + \frac{k}{m_s})} \tag{7}$$

where

$$\gamma = -\frac{f_2}{m_s} + g - \frac{k(L_0 + c)}{m_s} \tag{8}$$

$$\beta = \dot{x_2}(0) + x_2(0)\frac{b_2}{m_s} \tag{9}$$

$$\lambda_2 = -\frac{\gamma_2 u}{m_{eff}} - \frac{kL_0}{m_{eff}} - f_1 \tag{10}$$

Substituting (6) into (7), it can be shown after some algebra that

$$X_2(s) = \frac{x_2(0)s^4 + H_4 s^3 + H_3 s^2 + H_2 s + H_1}{s^2(s+a)(s+b)(s+c)} \tag{11}$$

where a, b, and c are roots to the polynomial

$$s^3 + s^2(\frac{b_2}{m_s} + \frac{b_1}{m_{eff}}) + s(\frac{k}{m_s} + \frac{k}{m_{eff}} + \frac{b_1 b_2}{m_{eff} m_s}) + \frac{k(b_1 + b_2)}{m_{eff} m s}$$

and

$$H_4 = x_2(0)\frac{b_1}{m_{eff}} + \beta \tag{12}$$

$$H_3 = x_2(0)\frac{k}{m_{eff}} + \frac{b_1\beta}{m_{eff}} - \gamma + \frac{kx_1(0)}{m_s} \tag{13}$$

$$H_2 = (\dot{x_1}(0) + \frac{x_1(0)b_1}{b_{eff}})\frac{k}{m_s} + \frac{k\beta}{m_{eff}} + \frac{b_1\gamma}{m_1} \tag{14}$$

$$H_1 = \frac{-\gamma k}{m_{eff}} - \frac{uk\gamma_2}{m_s m_{eff}} - (\frac{L_0 k}{m_{eff}} + f_1)\frac{k}{m_s} \tag{15}$$

Using (11), the Laplace transform for $X_1(s)$ can be re-written as

$$X_1(s) = \frac{s^2 x_1(0) + s(\dot{x_1}(0) + \frac{x_1(0)b_1}{m_{eff}}) + \lambda_2}{s(s+d)(s+e)} + Z(s) \tag{16}$$

where d and e are roots to the denominator of (6), and

$$Z(s) = \frac{\frac{k}{m_{eff}}(x_2(0)s^4 + H_4 s^3 + H_3 s^2 + H_2 s + H_1)}{s^2(s+a)(s+b)(s+c)(s+d)(s+e)} \tag{17}$$

Finally, taking the Inverse Laplace Transform of (11) and (16) with (17), solutions for all states can be easily found as

$$x_2(t) = A + Bt + Ce^{-at} + De^{-bt} + Ee^{-ct} \tag{18}$$

$$\dot{x_2}(t) = B - aCe^{-at} - bDe^{-bt} - cEe^{-ct} \tag{19}$$

$$x_1(t) = \hat{A} + \hat{B}t + \hat{C}e^{-at} + \hat{D}e^{-bt} + \hat{E}e^{-ct} + \hat{F}e^{-dt} + \hat{G}e^{-et} \tag{20}$$

$$\dot{x_1}(t) = \hat{B} - a\hat{C}e^{-at} - b\hat{D}e^{-bt} - c\hat{E}e^{-ct} - d\hat{F}e^{-dt} - e\hat{G}e^{-et} \tag{21}$$

Where A- E and $\hat{A}$- $\hat{G}$ are coefficients determined from partial fraction decomposition. Note that this analytic solution also agrees with our numerical solutions of the equations of motion, toward verifying accuracy.

D. Analytical Solution of Energy

Having access to analytic solutions as a function of time, we can gain more insight on the system by further considering the total energy. Recall from the compression phase of a real actuator shown in Fig 3 that when the stance phase had ended, x_1 is not fully compressed. That is, the actuator dynamics result in a higher-order system, where velocity of the body at take-off does not fully characterize the state nor the total energy. Due to this effect, attempting to characterize apex heights based solely on spring compression is difficult and not robust. However, considering the energy of the passive state over time allows us to exactly determine the resulting apex height for a given current input. Considering the system energy over time for an ideal hopper model without any loss terms involves calculating only the potential and kinetic energy over time. This overly-simplified case is shown in Fig. 12 where, as expected, the system energy remains constant over all time. This idealized assumption will not match real system behavior, however.

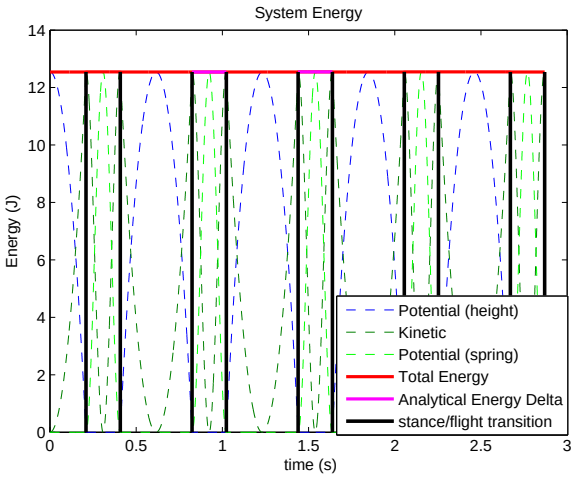

Fig. 12. System energy for a static drop simulation with an ideal hopper model.

We can determine the system energy more accurately for the real hopper model by using our improved model and analytic equations. The energy of x_2 is given by

$$U_{x_2} = \int m_s \ddot{x}_2 \, \mathrm{d}x_2. \tag{22}$$

Using 4, this can be expressed as

$$U_{x_2} = \int (k(-x_2 + L_0 + c) - m_s g)\dot{x_2} \, \mathrm{d}t + U_{delta} \tag{23}$$

where

$$U_{delta} = f_2(L_0 - x_2) + \int b_2 \dot{x_2}^2 \, \mathrm{d}t + \int k x_1 \dot{x_2} \, \mathrm{d}t \tag{24}$$

Using the analytical solutions (18) - (21), U_{delta} can be computed exactly over all time; it represents the sum of all energy loss terms, along with the energy added by the actuator. With these equations, we can calculate exactly what apex heights the system model will reach over all time,

given initial conditions entering the first stance phase. The system energy when undergoing a static drop test is shown in Fig. 13, where the Analytical Energy Delta (AED) (solid magenta) is precomputed using (24), and correctly equals the sum of kinetic and potential energy over time. Both the friction terms and the unsprung mass at the foot account for significant energy loss at each successive hop, and thus actuation is needed to introduce additional energy into the system, for either stochastic, height-varying terrain or for steady-state hopping or flat ground.

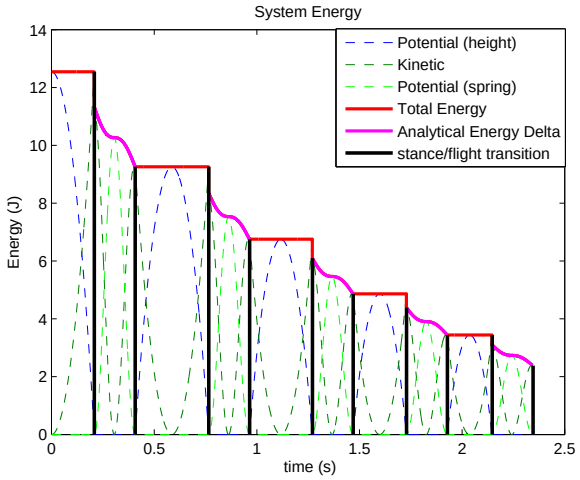

Fig. 13. System energy for a static drop simulation with a real hopper model. An instantaneous energy drop occurs at impact, due to the inelastic collision of the unsprung foot mass, and friction and damping during the continuous stance phase is even more significant.

Results for the same initial condition but with 3 Amps of actuation applied are shown in Fig. 14. As expected, the resulting energy levels are higher, and the system eventually converges to a steady-state hopping height. To maintain the energy level of the original hopper height, we can of course use our equations for the AED to find the magnitude of current needed to achieve this. From a controls perspective, this is obviously useful because it accurately characterizes the effect of the actuator on the resulting apex height of the system in a feedforward (and/or model predictive) sense.

IV. Energy-based Feedforward Control

As we have shown earlier, there is a strong desire to have feedforward control for this system, and having derived the analytical equations of system energy, this is simple to achieve. Using the analytic equation for the AED seen in (24), the system energy at the end of the stance phase, including all loss and additive terms, can be characterized as a function of apex heights. This is the basis for our feedforward controller and requires two assumptions. Assumption A: The energy of the system can be analytically computed, given an input current step. This assumption is valid, as we have previously shown. Assumption B: Ground collisions can be detected, and therefore the height of disturbances on terrain are known at touchdown. For the case of Hopper C, this is accomplished by augmenting the leg with a Hall Effect

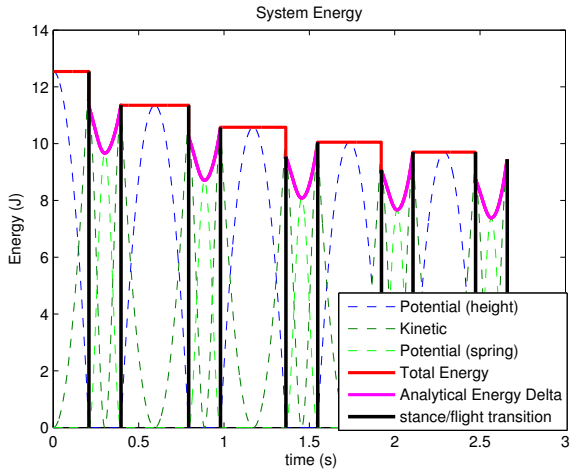

Fig. 14. System energy during an actuated simulation. Note energy is added via the U-shaded dips between the flat plateaus of the ballistic phase.

Sensor and a small magnet, and Fig 15 shows that ground impacts are easily detectable, and thus by recording apex height information the terrain level upon impact is known.

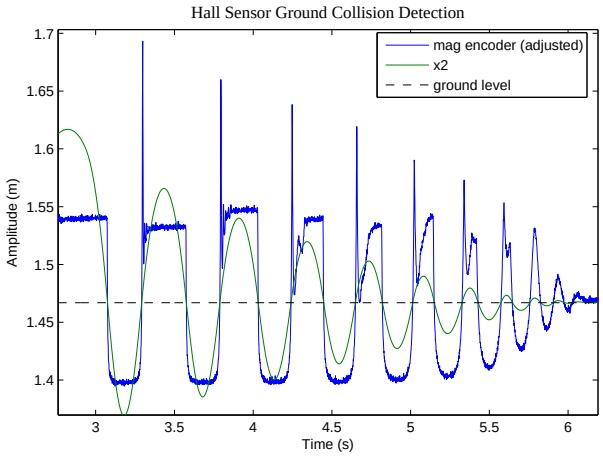

Fig. 15. Real-world sensor data of ground contact includes some noise effects. Hopper C has the capability to detect ground collisions using a Hall Effect sensor.

This control law is simple and easy to implement, as opposed to the Hopper B control method, which required significant system data and hand-tuning. At touchdown, the energy level of the system is computed by

$$E_{td} = \frac{1}{2} m_s v_{td}^2 \tag{25}$$

If a height disturbance is present, the disturbance energy is computed as

$$E_{dist} = \frac{1}{2} m_s g |x_{2ground} - x_2| \tag{26}$$

The AED is then computed for the end of the stance phase as $E_{delta}(u)$, and the controller selects current step magnitude u to minimize

$$J = |E_{delta}(u) - (m_s g h_{des} - E_{td} - E_{dist})| \tag{27}$$

where h_{des} is the desired next apex height. During the flight phase, a simple PD controller is used to return x_1 to the origin as "gently" as possible, 50ms before the start of the next stance phase. Results for this control strategy are shown for even terrain on Fig. 16, and rough terrain on Fig. 17. This feedforward control strategy performs well and is a significant improvement in controlling our hoppers.

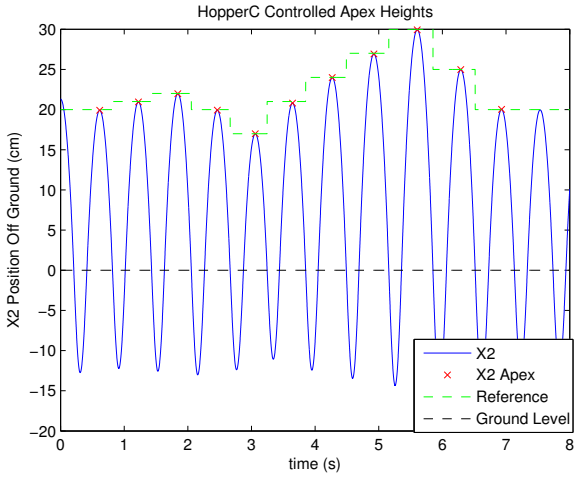

Fig. 16. Simulation results for the described energy-based controller on even terrain.

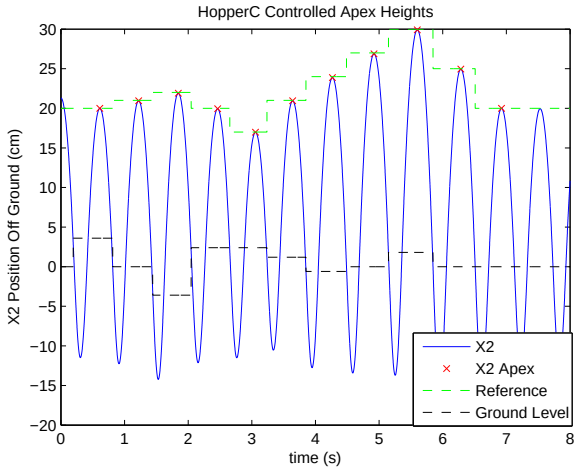

Fig. 17. Simulation results for the described energy-based controller on uneven terrain. Accuracy for even vs. uneven terrain is essentially identical.

V. Conclusion

We present a modeling technique motivated by actual hardware for a 1D series-elastic hopper to achieve accurate state tracking on uneven terrain. For practical hopping robots, actuators have real limits that must be modeled for state tracking to work well with feedforward control methods. For closed-form approximations of step-to-step dynamics, we argue such models are essential for both higher-level planning and low-level feedforward control.

The inclusion of both viscous and Coulombic friction in our hopper model increases model accuracy importantly and

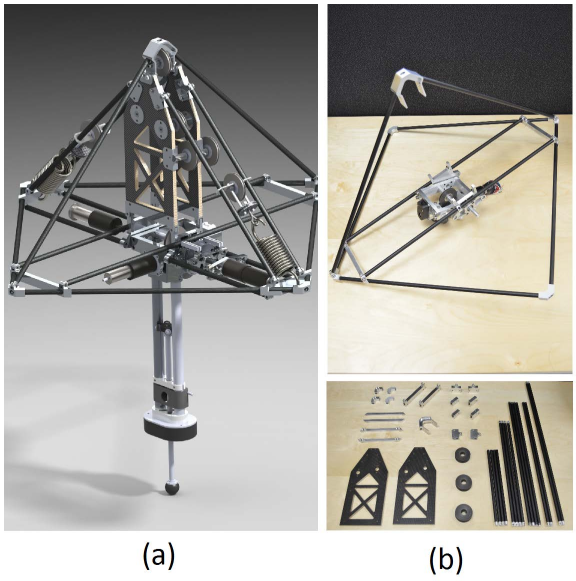

Fig. 18. Future work includes development of 2D hopper CAD design render in (a), with current progress of construction in (b).

improves control results. Achieving good tracking requires accurate forward model, which for this system is provided by complete analytic expressions. Analytical solutions are desirable for fast computational speed and are shown to exist and be correct under reasonable assumptions for this system. We also present a simple, energy-based feedforward controller that can accurately track references on rough terrain. Future work will consider applying these analytic expressions to approximate and control the dynamics of a 2D hopper currently under construction, seen in Fig. 18. Ultimately, having a good, closed-form model for a real-world, series-elastic, vertically constrained hopper is an important step toward future development of robust and energy efficient 2D and 3D hopping robots.

References

[1] R. McN. Alexander, "Three Uses for Springs in Legged Locomotion" in *The International Journal of Robotics Research*, vol. 9, no. 2, pp. 53–61, 1990.

[2] J. Schmitt and J. Clark, "Modeling posture-dependent Leg actuation in sagittal plane locomotion", Bioinspiration and Biomimetics, vol. 4, pp. 1–17, 2009.

[3] Ömür Arslan, Uluç Saranlı, Ömer Morgül, "Reactive Footstep Planning for a Planar Spring Mass Hopper," in *Proc. IEEE/RSJ Int. Conf. on Intel. Robots and Systems (IROS)*, pp. 160–166, 2009.

[4] M. H. Raibert, M. Chepponis, and H. Brown, "Running on Four Legs As Though They were One," *IEEE Journal of Robotics and Automation*, vol. 2, no. 2, pp. 70–82, 1986.

[5] Marc H. Raibert, "Legged Robots that Balance", MIT Press, 1986.

[6] H. Geyer, A. Seyfarth,and R. Blickhan, "Spring-mass running: simple approximate solution and application to gait stability," *Journal of Theoretical Biology*, vol. 232, pp. 315–328, Feb. 2005.

[7] W. J. Schwind and D. E. Koditschek, "Approximating the Stance Map of a 2 DOF Monoped Runner," *Journal of Nonlinear Science*, vol. 10, no. 5, pp. 533–588, 2000.

[8] U. Saranlı, O. Arslan, M. Ankarali, and O. Morgül, "Approximate analytic solutions to the non-symmetric stance trajectories of the passive spring-loaded inverted pendulum with damping," *Nonlinear Dynamics*, vol. 64, no. 4, pp. 729–742, 2010.

[9] J. Hodgins and M. Raibert, "Adjusting Step Length for Rough Terrain Locomotion," *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 289–298, June 1991.

[10] K. Byl, M. Byl, M. Rutschmann, B. Satzinger, Lous van Blarigan, G. Piovan, Jason Cortell, "Series-Elastic Actuation Prototype for Rough Terrain Hopping," in *Proc. IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, pp. 103–110, 2012.

[11] G. Piovan and K. Byl, "Two-Element Control for the Active SLIP Model," accepted in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2013.

[12] G. Piovan and K. Byl, "Enforced Symmetry of the Stance Phase for the Spring-Loaded Inverted Pendulum," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1908-1914, 2012.