Energy-Based Feed-Forward and Extended Feedback Linearization Control Strategies for Realistic Series Elastic Actuated Hopping Robots

Pat Terry and Katie Byl

***Abstract*— In this paper, we describe modeling and control techniques for a real-world series-elastic actuated 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. However, real-world systems cannot be fully 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. Implementing feedback based control for series-elastic actuators is difficult when the actuator has real dynamics, as the input variable does not instantaneously change the leg length acceleration. In addition to providing analytically exact feed-forward based control, we also provide a direct method for regulating apex height by extending the method of feedback linearization to a higher order. Motivated by actual hardware, our work here addresses modeling and control 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.**

I. Introduction

Hopping robots, aka hoppers, are useful systems 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 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. Two control implementations are considered: an analytically exact feed-forward energy-based solution, and a feedback method based on an extension of Feedback Linearization to a higher order.

Using compliant springs as energy storage devices for legged locomotion is a concept that has been extensively studied [1], and has led 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 led to the development of many hopping and other leg-actuated robots, including well-known work led by Raibert [5] which produced many successful experimental results. In contrast to Raibert's work, our control strategies aim to minimize hand-tuned experimental parameters in an effort to increase robustness on unknown terrain environments. 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. Another important requirement 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 [10], [11], toward computationally fast planning with a limited, noisy looka-

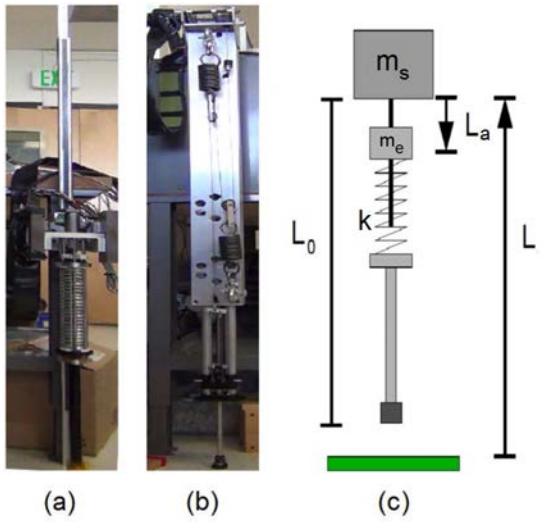

Fig. 1. (a) Hopper B, (b) Hopper C, and (c) schematic for hopper model, which is essentially a large leg, where L represents the length from the ground to the bottom of the leg with natural length L_0. The system is actuated by a series elastic actuator L_a, which compresses the spring from the opposite end.

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

head 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 (2D and 3D) real-world hopper dynamics significantly. Second, accurate planning for 2D and 3D hoppers requires good feed-forward control; since contact with the ground is brief (e.g., less than 0.25 sec), control should potentially use elements of both feed-forward and feedback, however implementing feedback controllers for series-elastic actuators as a function of desired leg compression is a challenging problem due to the inability to instantaneously control the leg acceleration, which is set by the length of the series spring element.

We have developed two vertically constrained hoppers [12] and a boom-constrained 2D hopper. Fig. 1 shows the two vertical hoppers, Hopper B and Hopper C, along with the schematic used for modeling. Each robot has an actuator in series with a spring to allow for the addition of energy in the stance phase. The actuators in these robots have dynamics, therefore 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. 2, where in fact the actuator is still undergoing compression when the stance phase has ended. Additionally, simply including only linear (viscous) damping terms is not sufficient; non-linear Coulomb frictional effects must be estimated as well to have an accurate model.

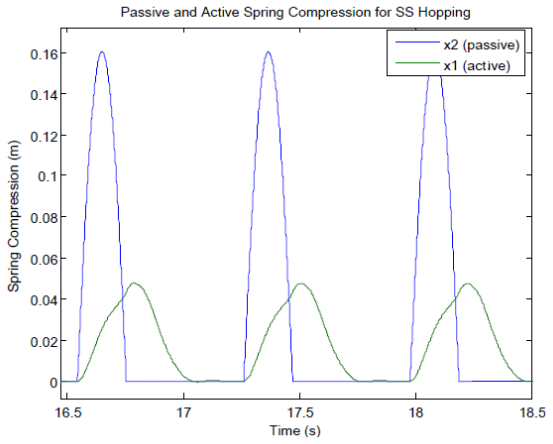

Fig. 2. Compression cycles during stance phase for steady-state operation of the Hopper C hardware. In the data above, $x_1 = L_a$ and $x_2 = max((L_0 - L), 0)$, which represents spring compression. Note that actuator dynamics happen at a time scale similar to that of hopper dynamics and cannot be well-approximated as instantaneous.

The rest of the paper is organized as follows. In Section II we define the dynamics of the system, followed by Section III which presents two control algorithms for regulating desired apex heights on rough terrain: the first based on feed-forward control, and a second method extending Feedback Linearization to the jounce of the leg. Section IV presents results for a simulation study considering apex tracking robustness of the two methods under the assumption that identified frictional system parameters used by the controllers are incorrect. Lastly, Section V provides an extension to 2D hopping systems.

II. System Dynamics

In this section we introduce the model for our 1D hopper system seen in Fig. 1. The system can be described as having a *flight phase* and a *stance phase*. As real world actuators have significant sources of both linear and nonlinear friction, we choose to model our system's losses using both linear damping and non-linear Coulomb friction. It can be shown using a Lagrangian approach that the dynamics of the stance phase are

$$G_1 = \ddot{L}_a = \frac{1}{m_e}(k(L - L_a - L_0 - c) - b_1\dot{L}_a - \gamma u) - f_1 sign(\dot{L}_a) \quad (1)$$

$$G_2 = \ddot{L} = \frac{1}{m_s}(k(L_a - L + L_0 + c) - b_2\dot{L} - m_s g) - f_2 sign(\dot{L}) \quad (2)$$

The flight phase dynamics simply follow ballistic equations and are therefore not shown. L represents the vertical leg length, interpreted as spring compression during stance phase and vertical height during flight phase. L_a represents the series elastic actuator that moves positively to provide spring compression. The variables k, c, m_s, m_e, g represent the spring constant, spring pre-load, sprung mass, effective actuator mass, and gravity constant respectively. The loss coefficients of the system are b_1, b_2, f_1, f_2 and γ converts the input current u into input force to the actuated system.

System identification of our hardware resulted in parameters shown in Table I. Additionally, 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. 3. 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 L_a is intentionally limited to within 1-2cm of bottoming out experimentally. This particular measurement does not include using the actuator to actively remove energy due to hardware limitations. The new model correctly captures the behavior of the bounding line, although some error is still present, likely due to errors in the identified frictional terms or the assumed frictional behavior.

III. Control Techniques for Apex Regulation

This section presents two methods for regulating desired apex heights at each step, in the presence of small terrain disturbances. For this work, full state estimation is assumed, the collision with the ground is assumed to be instant, inelastic, and detectable at each step, and control of apex height is achieved through torque input to a series elastic actuator during the stance phase.

L_0	0.534 m	Natural leg length
m_e	7.11 kg	Eff. actuator mass
m_s	8.475 kg	Total sprung mass
m_f	0.95 kg	Unsprung mass
k	2,389 N/m	Spring constant
c	0.021 m	Spring pre-load
γ	51.1 N/A	Actuator constant
b_1	3.333 Ns/m	Act. lin. friction
f_1	1.52 N	Act. coulomb friction
b_2	1.18 Ns/m	L lin. friction
f_2	1.03 N	L coulomb friction
ω_n, ζ, p_3, p_4	10, 0.8, 75, 100	EFL gain parameters
$[f_{1,min}, f_{1,max}]$	[0, 1.75]	Iteration range 1
$[b_{1,min}, b_{1,max}]$	[0, 30]	Iteration range 2
u_{max}	20 Amps	Max allowed u
μ	0.05	Arctan constant

TABLE I

MODEL PARAMETERS USED FOR SIMULATION STUDY

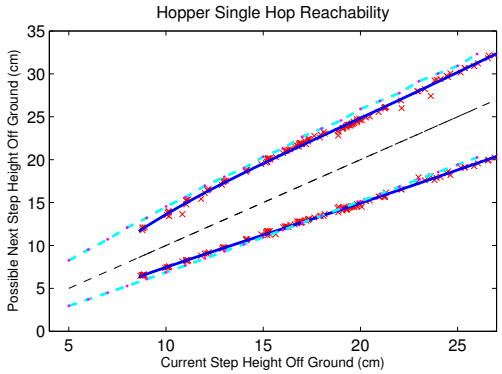

Fig. 3. The reachability curve shows at each current apex state (x-axis), the possible next apex states are bounded by two linear curves. The blue lines and red markers are experimental measurements, and the cyan dashed lines are calculated using our proposed model. The similarity between the two curves justifies studying this model to generate control strategies for our hardware, and the curve itself is useful for generating feasible trajectories.

A. Energy-Based Feed-Forward Control

One of the advantages of modeling a 1D Hopper is that an analytical solution exists and can be determined exactly. Thus, at touchdown a trajectory can be generated based on initial touchdown conditions that characterizes the motion throughout the stance phase of the system for all future time. The analytical solution presented here is generated for actuator inputs as constant current steps. Assuming that the control actuation is a step input at touchdown, we can form a piecewise linear analytic solution for the dynamics during stance. Each time either L or L_a changes sign the equations must be broken into an additional piece, and the remaining trajectory calculated from new initial conditions. The i^{th} piece of the trajectory, for $t \in [t_{i-1}, t_i]$, can be calculated as

$$L(t,i) = A + Bt + Ce^{-at} + De^{-bt} + Ee^{-ct} \quad (3)$$

$$\dot{L}(t,i) = B - aCe^{-at} - bDe^{-bt} - cEe^{-ct} \quad (4)$$

$$L_a(t,i) = \hat{A}+\hat{B}t+\hat{C}e^{-at}+\hat{D}e^{-bt}+\hat{E}e^{-ct}+\hat{F}e^{-dt}+\hat{G}e^{-et} \quad (5)$$

$$\dot{L_a}(t,i) = \hat{B}-a\hat{C}e^{-at}-b\hat{D}e^{-bt}-c\hat{E}e^{-ct}-d\hat{F}e^{-dt}-e\hat{G}e^{-et} \quad (6)$$

where all the various coefficients, A ,B,C,D,E; $\hat{A}$,...,$\hat{G}$, and a,...,e are determined from partial fraction decomposition and are constant multiples for the i^{th} initial condition. Often the sign of $\dot{L}$ only changes once and the sign of $\dot{L}_a$ will remain constant throughout the stance phase, thus for many cases during typical hopping the equations only need be broken into two pieces.

To construct a feed-forward controller towards the purpose of regulating apex heights, we consider the total energy of the system. The energy of the spring-mass system is given by

$$U_L = \int (k(-L + L_0 + c) - m_s g)\ddot{L}\,\mathrm{d}t + U_{delta} \quad (7)$$

where

$$U_{delta} = f_2(L_0 - L) + \int b_2\dot{L}^2\,\mathrm{d}t + \int kL_a\dot{L}\,\mathrm{d}t \quad (8)$$

Using the analytical solutions (3) - (6), 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.

Results for a simulation with 3 Amps of actuation applied are shown in Fig. 4. The Analytical Energy Delta (AED) (solid magenta) is precomputed using (8), 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. To achieve consistent steady-state hopping, we can of course use our equations for the AED to find the magnitude of current needed to achieve this.

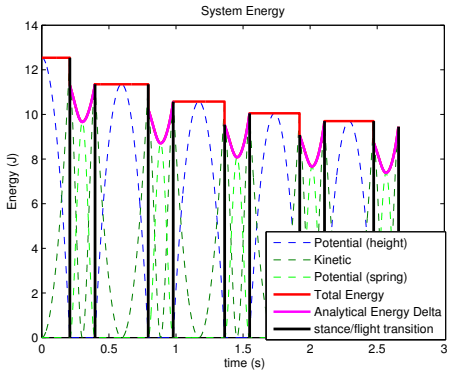

Fig. 4. System energy during an actuated simulation. Note energy is added via the U-shaded dips between the flat plateaus of the ballistic phase.

Using the analytic equation for the AED seen in (8), 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 feed-forward (FF)

controller, which is simple to implement. At the instant after touchdown, the energy level of the system is computed by

$$E_{td} = \frac{1}{2} m_s \dot{L}^2 \tag{9}$$

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

$$E_{dist} = \frac{1}{2} m_s g |y_{ground} - L| \tag{10}$$

The AED is then computed for the end of the stance phase as $E_{delta}(u_{FF})$, and the controller selects current step magnitude u_{FF} to achieve the correct next-apex energy level and therefore minimize

$$J = |E_{delta}(u_{FF}) - (m_s g h_{des} - E_{td} - E_{dist})| \tag{11}$$

where h_{des} is the desired next apex height. Although the above cost function may not be analytically solvable, finding the correct value of u_{ff} can be accomplished by simply calculating $E_{delta}(u_{FF})$ over a window of values, since the analytic equations can be calculated quickly, and then simply selecting the value which yields the smallest observed J.

In our implementation for computing u_{FF}, the solution to Eq. 11 is found via a binary search algorithm using the analytic equations, with minimum and maximum search values being actuator current limits, 0 to 20 Amps. The average computation time of the FF solution for the results presented in this paper on a modern computer is 18.5ms. This is a significant fraction of the stance phase time, but FF computation can be accomplished during the descending portion of the flight phase and/or pre-computed into look-up tables for real-time control implementation.

Results for a simulation implementation of this algorithm as described are shown in Fig. 7, and are quite reasonable when the system parameters are known correctly. It is of interest to note that in addition to solving for current step magnitude u_{ff}, trajectories for all states are also produced via the analytical solution and can be implemented as references for a feedback controller in real time, which is explored in the next section.

B. Extended Feedback Linearization

This section discusses a more direct feedback method for regulating apex heights. The reasons to develop such a method are (1) typically using only feed-forward methods will result in significant errors from parameter mismatch and (2) exact analytical solutions for more general 2D hopping systems are sometimes impossible to calculate and difficult to estimate. Therefore, we desire a control strategy that lets us directly control the error dynamics of L using u. This entails having a feedback controller that enforces references on $\dot{L}$ such that its value at take-off results in a desired energy level.

We start by considering the equation for the stance dynamics of the leg length, seen in Eq. 2. We would like to use u to directly control L (not L_a) as it is our actuator input variable. Since our system has nonlinear frictional effects we would like to negate, we propose that using feedback linearization can accomplish both goals. Since u does not directly effect $\ddot{L}$ due to the presence of the series elastic spring element, we must map the feedback linearization to a higher order state. In this case, we must take two additional derivatives for our control variable to directly effect L, therefore the relative degree of our system with L as the chosen output is 4. We denote this Extended Feedback Linearization (EFL) for shorthand simply because the order of our feedback linearizing controller is not 2, which is standard practice for classical implementations of PFL on underactuated mechanical systems [13] [14]. Since the Coulombic friction term is not differentiable, we approximate it with a scaled $\arctan$ function, which is reasonably close when μ is sufficiently small. We construct the approximate 4th derivative of L, aka the jounce (or snap), as:

$$\begin{aligned}
\tilde{G}_2 &= \frac{1}{m_s}(k(L_a - L + L_0 + c) - b_2\dot{L} - m_s g) - \frac{2f_2}{\pi} atan(\frac{\dot{L}}{\mu}) \\
\dddot{L} &\approx \frac{d}{dt}\tilde{G}_2 = -\frac{k}{m_s}(\dot{L} - \dot{L}_a) - \tilde{G}_2(\frac{b_2}{m_s} + \frac{2f_2}{\pi\sqrt{\mu}}\frac{1}{\mu + \dot{L}^2}) \\
\ddddot{L} &\approx \frac{d}{dt}(\frac{d}{dt}\tilde{G}_2) = \beta_L(X)u + \epsilon_L(X)
\end{aligned} \tag{12}$$

where β_L and ϵ_L are functions of the state $X = [L_a\ L\ \dot{L}_a\ \dot{L}]$ and G_2 represents the dynamics of L introduced in Eq. 2. Next we define our control law as:

$$u_{EFL} = \frac{1}{\beta_L}(-\epsilon_L + v_L). \tag{13}$$

In other words, u cancels the natural dynamics (given by ϵ_L) and forces any errors in L to decay via linear, fourth-order dynamics that we set through v_L. Specifically, we choose:

$$\begin{aligned}
v_L &= v_1 + v_2 \\
v_1 &= K_p(L_{ref} - L) + K_d(\dot{L}_{ref} - \dot{L}) \\
v_2 &= K_{dd}(\ddot{L}_{ref} - \ddot{L}) + K_{ddd}(\dddot{L}_{ref} - \dddot{L}).
\end{aligned} \tag{14}$$

Since we increased the order of the derivative the feedback linearization is acting on by two, we now require four total poles in the closed-loop dynamics of $L(t)$. This in itself is an open problem with many solutions. One option for setting the controller gains is to first select a dominant pole-pair with natural frequency ω_n and damping ratio ζ, and then set a significantly faster decay rate for the two remaining, real-valued poles, p_3 and p_4. For a chosen set of w_n, ζ, p_3, p_4, the gains are:

$$\begin{aligned}
K_p &= p_3 p_4 w_n^2 \\
K_d &= (p_3 + p_4)w_n^2 + 2p_3 p_4 \zeta w_n \\
K_{dd} &= w_n^2 + 2\zeta(p_3 + p_4)w_n + p_3 p_4 \\
K_{ddd} &= p_3 + p_4 + 2\zeta w_n
\end{aligned} \tag{15}$$

Additionally, we require both references and estimates of the acceleration and jerk of the system, which can be calculating by using Eqs. 2 and 12. Utilizing this control method for the purpose of apex regulation involves designing compatible stance phase trajectories such that the terminating value of $\dot{L}$, e.g. the take-off velocity, is set corresponding

to a desired energy level similar to that of Eq. 9. Example trajectories and tracking performance are shown in Fig. 5, with the actuator output show in Fig. 6.

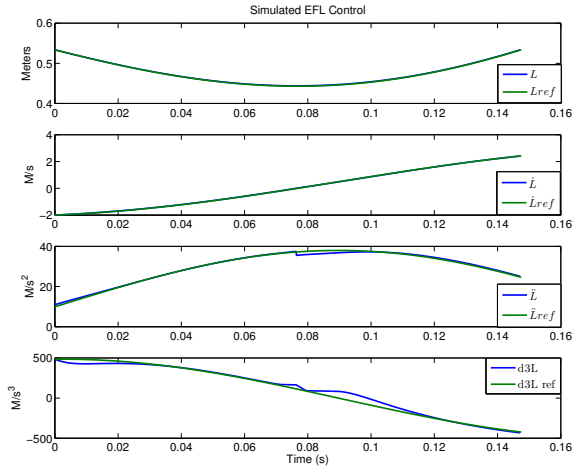

Fig. 5. Simulation results for the described EFL controller, showing the controller's ability to track references on L using the series elastic actuator L_a with reasonable error. The discontinuity in acceleration is due to the Coulombic friction term. Trajectories were designed via the simulation of an equivalent system with the Coulombic friction removed, with an input current step of 6 amps. This was done simply to show the controller's ability to track smooth trajectories, even with modeling error.

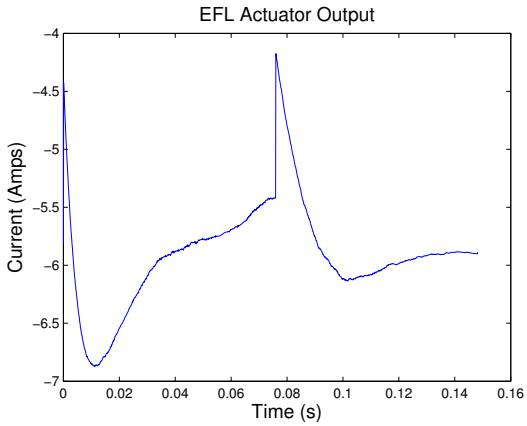

Fig. 6. Actuation requirement in Amps for the described EFL controller simulation. Compared to the actuator effort of the FF strategy for identical trajectories, which is a constant 6 amps, the maximum required current is slightly higher while the average current usage is only 5.94 amps.

We lastly consider the intuitive idea of simply combining the feed-forward method with the feedback method, thus we construct a new control law that combines both controllers seen in Eqs. 11 and 13 as:

$$u = u_{FF} + u_{EFL} \tag{16}$$

This control strategy is simply an initial feed-forward constant combined with feedback provided by the EFL controller. The trajectories required by the EFL controller can be generated on-the-fly by simply using the solutions of Equations 3- 6, which are calculated by the FF controller.

IV. Simulation Robustness Study

Often when performing system identification of real systems, the frictional terms can be the most challenging to determine precisely since they can vary both over time and the exact system state. This has proven to be a significant practical challenge in our laboratory hardware, for example. Therefore, it is of particular interest to study control results when the parameters used by the controller for the actuated system's frictional parameters, e.g. f_1 and b_1 in Eq. 1, do not match those of the real system. In this study, we assume the true dynamics of the system are those presented in Section II, and vary the frictional parameters used by the controller in order to study how these errors affect performance of apex tracking.

We use a base trajectory of apex heights we would like the system to follow for these studies. The apex heights were selected to span a reasonable amount of the reachable states seen in Fig. 3, and the ground level is also randomly varied on a step-to-step basis in order to simulate minor terrain variations. Simulation results for both control methods are shown in Figures 7 and 8. With correct parameters, the FF method has negligible error, while the EFL method has some small tracking error. To conduct the simulation study, we define a range of controller frictional values of the series elastic actuator to iterate over as $[f_{1,min}, f_{1,max}]$ and $[b_{1,min}, b_{1,max}]$, which are the values the controller will use on the real system. The controllers are commanded to track the base apex height trajectory for each parameter, and the sum of squared error (SSE), normalized by the square of the reference to be unitless, is recorded by summing the ratio of squared apex height clearance off ground errors to every desired apex in each trial. For reference, the SSE of the FF implementation using correct parameters is 0.12, and the average apex error (AE) was 0.7%. In contrast, the EFL has SSE 2.13 and AE 3.8% with correct parameters. The parameters used to conduct the simulation study are shown in Table I.

Results for the simulation study are shown in Figures 9 - 11. As expected, the feed-forward controller performs poorly when the controller parameters do not match the system dynamics. While the minimum error is indeed when using the true parameters, other points with reasonably small error exist and represent cases when the friction and damping terms are identified incorrectly, but the total sum of their effect on energy loss is approximately the same. One example of this is highlighted as a red dot on Fig. 10, in which case the controller uses $f_1 = 0$ and $b_1 = 7.346$, meaning the model is assumed by the controller to have only linear damping terms.

The EFL implementation performs significantly better under parameter variation. In fact, the SSE only increases by roughly 40% over the full iteration range, however the minimum error is unacceptably large. Results for the total combined FF and EFL strategy are seen in Fig. 11, and indeed this control strategy does capture the positive qualities of both methods, exhibiting a quite reasonable base SSE

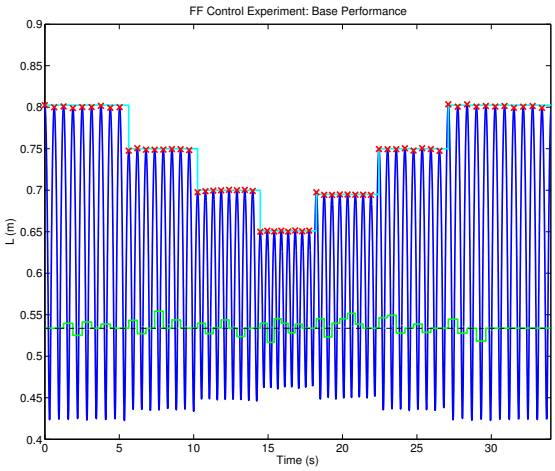

Fig. 7. The apex tracking accuracy shown here is the "base" performance for the FF method, with the controller using correct parameters that match the system dynamics. The cyan line shows the desired references, with achieved apexes marked by red X's, with the terrain level plotted on top of the leg's natural length L_0 shown in green.

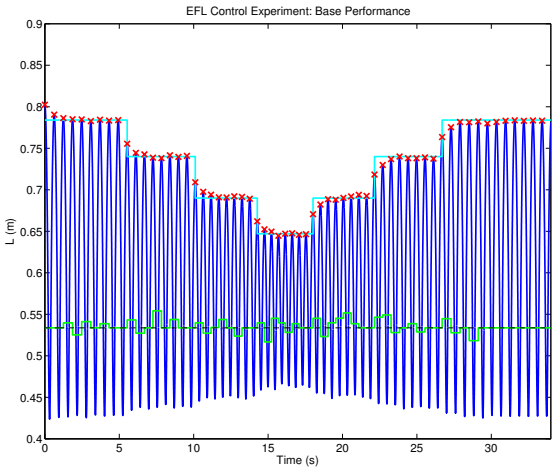

Fig. 8. The apex tracking accuracy shown here is the "base" performance for the EFL method, with the controller using correct parameters that match the system dynamics. With perfect parameter identification, the accuracy of this method is slightly less than the FF strategy, although some accuracy could likely be improved by controller gain and trajectory optimization.

of 0.33, which translates to an average apex error (AE) of approximately 1%, and maintaining fairly good performance under parameter variation.

V. Extension To 2D Hopping Systems

The control methods presented thus far have concerned apex regulation for vertically constrained 1D hopping robots, and we now consider a very simple algorithm for applying these control methods towards 2D hopping. The 2D Hopper being developed in our lab is shown in Fig. 12, and while mechanically different from classical Raibert hopping robots [5], the dynamics can be represented in a very similar model, therefore the full model derivation for dynamics of our 2D

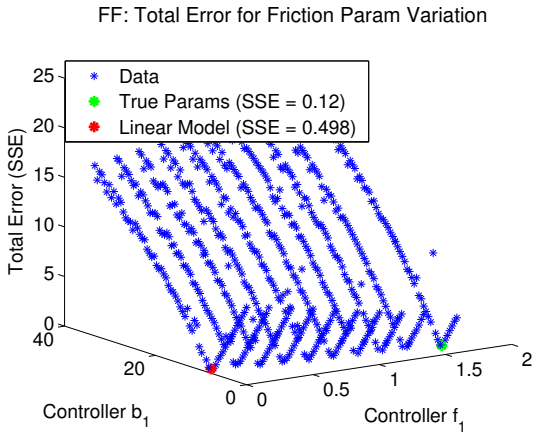

Fig. 9. Results for variation of the actuator's (L_a) frictional terms used by the FF controller. These results have two important features (1) the feed-forward method is quite sensitive to incorrect frictional parameters and (2) for apex regulation, attempting the approximate the system with only linear frictional terms results in roughly five times more total error.

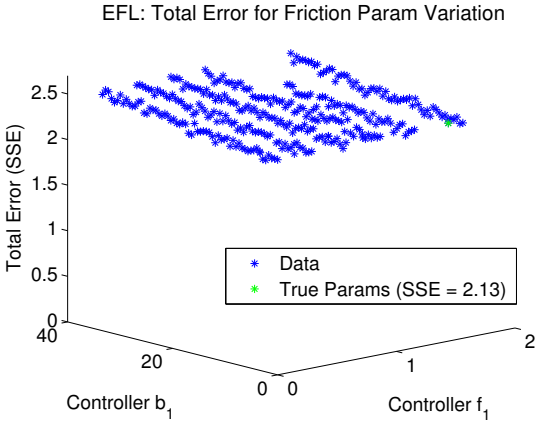

Fig. 10. Results for variation of the actuator's (L_a) frictional terms used by the EFL controller. Compared to the results of the FF implementation we make two observations (1) the surface is significantly less structured due to the feedback gains and (2) while the EFL performance using the correct parameters is worse, the controller's robustness to the variation of these frictional parameters is significantly better than that of the FF controller.

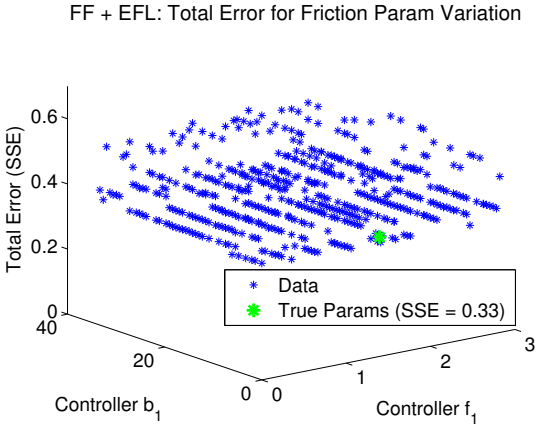

Fig. 11. Results for variation of the identified actuator (La) for the proposed combined FF and EFL control strategy. This strategy maintains the robustness to frictional parameter variation of the EFL strategy while having significantly improved base performance.

simulation work is omitted from this section due to space limitations. The full dynamics for our system consist of both 1D states L and L_a, in addition to a leg angle θ with leg mass and inertia m_l and I_l, and a body angle ϕ with body mass m_s. However, since our 2D Hopper hardware is constrained by a boom, we have constructed a mechanical lock to prevent body rotation, which allows us to study models more akin to SLIP but with realistic hardware elements. Body rotation will be enabled for future work. Therefore, for this work we consider the states to simply be L, L_a, θ and their respective derivatives.

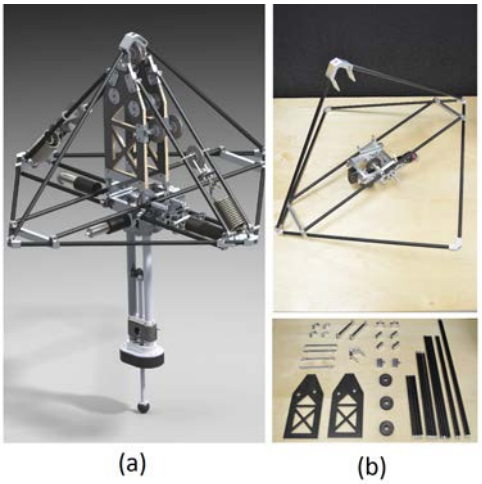

Fig. 12. Future work includes development and control of 2D hopping robot, with CAD design render in (a), and current progress of construction in (b). The leg mechanism is the same used in the 1D Hopper C. Although the robot has enough actuators for 3D hopping, the initial design includes a boom to constrain motion to 2D hopping, with an optional body lock.

We consider only the problem of apex regulation in this work to simply illustrate how the 1D control methods developed can be extended for use during 2D operation. Although the system has a leg actuator u_θ in addition to the series elastic actuator, for this work we only allow this actuator to position the leg in a desired touch-down angle during the flight phase. Only the series elastic actuator outputs power during the stance phase. To achieve simple forward hopping, during the flight phase the leg angle θ is set using a classical partial feedback linearizating controller to a desired touch-down angle r_θ as

$$r_\theta = \theta_0 + K(\dot{x}_r - \dot{x}_{hip}) \tag{17}$$

Where θ_0 is a constant touch-down angle, and $\dot{x}_{hip}$ is the forward velocity of the hip joint, with forward velocity reference $\dot{x}_r$ and gain K. Given this forward gait, we first naively attempt to simply use our combined FF and EFL control strategy of Eq. 16 to track a set of apex trajectories during forward movement. Simulation results with θ_0 and $\dot{x}_r$ of -4.5° and 0.66$\frac{m}{s}$ respectively are show in Fig. 13, where it is clear significant apex error is present for all time. However, due to the EFL controller component accuracy providing feedback on leg compression trajectories, we see that the controller error is very consistent and appears to simply be

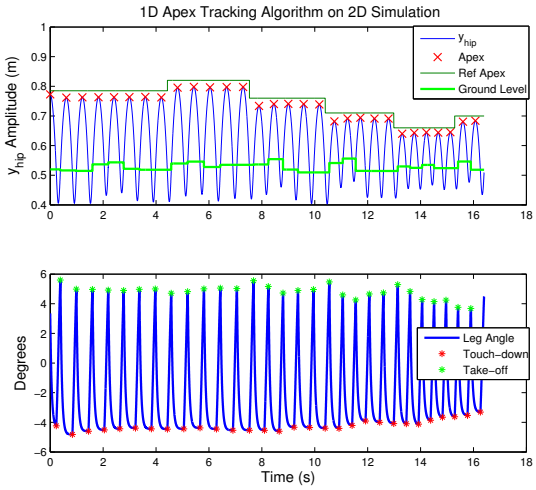

Fig. 13. Attempting to use the combined EFL and FF 1D apex regulation control strategy for our 2D hopper results in significant apex error. However, due to the EFL controller accurately enforcing trajectories for the leg length, the effect of our 2D system, i.e. having non-zero leg angle and velocity, can be viewed as an energy offset. The touch-down and take-off angles, while not exactly symmetric, are fairly close to one another for the simple 2D control strategy described.

an energy offset. This energy difference E_θ can in fact be calculated, and is represented as

$$\begin{aligned} E_\theta &= \frac{1}{2} m_s (\dot{y}_{TO})^2 - m_s g (h_{des} - L_0) \\ \dot{y}_{TO} &= (1-\rho)\dot{L}_{TO}\cos(\theta_{TO}) - \dot{\theta}_{TO} L_0 \sin(\theta_{TO}) \end{aligned} \tag{18}$$

Where $\dot{y}_{TO}$ is the take-off hip vertical velocity, h_{des} is the desired next apex height, and $\dot{L}_{TO}$, θ_{TO}, $\dot{\theta}_{TO}$ are the velocity states of the system at the end of the stance phase, right before take-off. The parameter ρ accounts for energy losses of the system at touch-down and take-off due to the unsprung mass of the leg and mechanical take-off mechanism. Therefore at touch-down, the next take-off state must be partly estimated in order to correct for the energy offset. We note however that since our EFL controller component is regulating L and $\dot{L}$, the take-off value $\dot{L}_{TO}$ will simply be the terminating value of the trajectory selected by the FF controller component as

$$(1-\rho)\dot{L}_{TO} = \sqrt{2g(h_{des} - L_0)} \tag{19}$$

We also note from Fig. 13 that during stance the angle of the leg swings to within a degree of the touch-down angle. For our initial 2D testing, we use the following simple but adequate approximation for the take-off values:

$$\begin{aligned} \theta_{TO} &\approx -\theta_{TD} \\ \dot{\theta}_{TO} &\approx \dot{\theta}_{TD} \end{aligned} \tag{20}$$

We can then use Eqs. 19 and 20 in Eq. 18 to approximate the energy difference as $\tilde{E}_\theta$, and the only change to our control algorithm is in the FF energy cost function as

$$J_{2D} = |E_{delta}(u_{FF}) - (m_s g h_{des} - E_{td} - E_{dist} + \tilde{E}_\theta)| \tag{21}$$

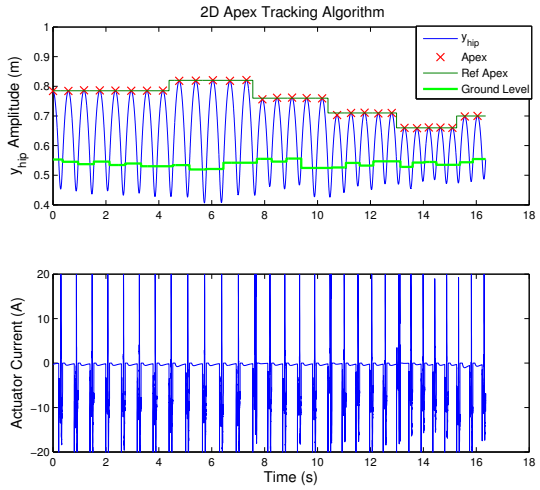

Fig. 14. Estimating and correcting the energy offset results in good performance even in the presence of rough terrain, similar to the 1D results. The actuator output, while exhibiting some initial saturation at touch-down, is reasonably well behaved. The average current consumption for the above simulation is 2.63 Amps, which is well within continuous current limits of hardware.

Thus, using the new value for u_{FF} and new EFL trajectories generated by the FF solution, we implement the controller exactly as in the 1D case. Results for this algorithm are show in Fig. 14, and indeed the energy offset has been corrected and apex tracking performance is virtually identical to the 1D simulations. The simulation data provides compelling initial evidence that improved 1D control is also practical and applicable in improving performance for 2D hopping. It is also of interest to note that the additional leg angle actuator available on our robot's hardware was not active during the stance phase for this study, and could be used in conjunction with designed stance phase trajectories to enforce exact symmetry conditions of the leg angle, which has been previously studied [10].

VI. Conclusion

We have presented modeling and control techniques for a realistic series-elastic actuated hopping robot 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 feed-forward 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 feed-forward and feedback control.

This work focuses in a very deliberate way on application of classical control techniques toward improved control and predictive modeling of legged systems. We note there is often a disconnect between simplified (e.g., lossless) theory and real-world (hardware-limited) results within legged robotics. Often, research aims directly at one corner or the other in the field of locomotion control. Here, our aim is to improve both reliability (e.g., ability to recover from terrain perturbations) and agility (e.g., ability to accurately go to any of a family of reachable future states) of spring-legged robots.

The inclusion of both viscous damping and Coulombic friction in our hopper model increases model accuracy significantly and is supported by our control architecture. Achieving good tracking is enhanced by having an accurate forward model, which for this system is provided by complete analytic expressions. We provide a feedback linearization based control method about the jounce of the leg that can accurately track references directly on L and remain reasonably accurate on rough terrain, and provide simple examples as to how these control algorithms may be extended to 2D hopping systems. Future work includes demonstrating this control approach in both 2D and 3D in real-time on the prototype hardware, investigating the use of additional leg angle actuators during the stance phase, incorporating alternate actuation trajectories to reduce energy consumption, and feed-forward apex height planning to accommodate rougher terrain.

Acknowledgments

This work is supported in part by DARPA's M3 program (Grant No. W911NF-11-1-0077) and by the Army Robotics CTA (Grant No. W911NF-08-R-0012).

References

[1] R. M. Alexander, "Three uses for springs in legged locomotion," *Int. 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, no. 1, pp. 1–17, 2009.

[3] Ö. Arslan and U. Saranlı, "Reactive planning and control of planar spring-mass running on rough terrain," *IEEE Trans. on Robotics*, vol. 28, no. 3, pp. 567–579, 2012.

[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] M. H. Raibert, *Legged Robots that Balance*. Cambridge, Massachusetts: MIT Press Series in Artificial Intelligence, 1986.

[6] a. R. B. H. Geyer, A. Seyfarth, "Spring-mass running: simple approximate solution and application to gait stability," *Journal of Theoretical Biology*, vol. 232, no. 1, pp. 315–328, 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] M. A. U. Saranlı, O. Arslan 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, no. 1, pp. 289–298, 1991.

[10] G. Piovan and K. Byl, "Enforced symmetry of the stance phase for the spring-loaded inverted pendulum," in *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 1908–1914, 2012.

[11] G. Piovan and K. Byl, "Two-element control for the active slip model," in *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 5656–5662, 2013.

[12] K. Byl, M. Byl, M. Rutschmann, B. Satzinger, L. van Blarigan, G. Piovan, and J. Cortell, "Series-elastic actuation prototype for rough terrain hopping," in *Proc. IEEE Int. Conf. on Tech. for Practical Robot App. (TePRA)*, pp. 103–110, 2012.

[13] M. W. Spong, "Partial feedback linearization of underactuated mechanical systems," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 314–321, 1994.

[14] P. Terry and K. Byl, "A higher order partial feedback linearization based method for controlling an underactuated hopping robot with a compliant leg," in *Proc. IEEE Conf. on Decision and Control*, 2014.