

Towards Precise Control of Hoppers: Using High Order Partial Feedback Linearization to Control the Hopping Robot FRANK

Pat Terry, Giulia Piovan, and Katie Byl

Abstract—In this paper, we describe modeling and control techniques for series-elastic actuated hopping robots. There is an abundance of work regarding the implementation of highly simplified hopper models, the prevalent example being the SLIP model, 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. Additionally, implementing feedback control for real series-elastic actuators is difficult as the input variable does not instantaneously change the leg length acceleration. In this work we provide both hardware and simulation results that illustrate how high-order partial feedback linearization can be implemented directly on the leg length state and show how to apply these results so that algorithms in the SLIP literature can be accurately implemented in more realistic hopping systems to accomplish tasks such as apex state tracking and precise foothold planning.

I. INTRODUCTION

Among the different types of legged robots, hopping robots, aka hoppers, can be classified as one of the simplest sufficient models that capture the important features encompassed in dynamic locomotion: underactuation, compliance, and hybrid features. Studying the behavior of a single leg and extending this to multiple legs has been shown to be effective [1] and has led to the development of many hopping and other leg-actuated robots, including well-known work led by Raibert [2], which produced many successful experimental results. To achieve highly compliant legs capable of dynamic running gaits, we typically consider robot models with springs in series with their legs [3], [4]. Particularly, the Spring Loaded Inverted Pendulum (SLIP) model [5] is a simplistic model that is widely used to help describe the dynamics of many legged animals and legged robots and, most particularly, of hopping robots [6], [7]. This work aims to facilitate application of approximate, closed-form planning and control solutions for hopper models to real-world robots.

Many studies using SLIP models focus on the use of leg angle at touchdown and/or modification of spring force during ground contact as the only available control actions [8], [9]. Such works investigate control of step length and jumping height without regard for realistic actuator dynamics. One of the main difficulties in modeling a 2D

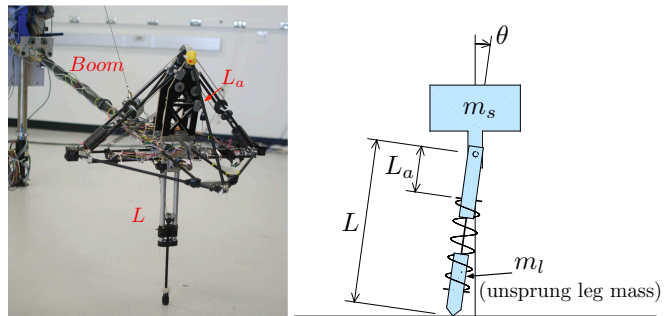


Fig. 1. The hopping robot FRANK has three actuators: A cable driven series elastic actuator (SEA) to inject energy into the system, a planar leg angle actuator to control the forward touch-down angle (θ), and a second leg angle (θ_{\perp}) actuator perpendicular to the direction of motion, to keep the foot from slipping during the stance phase. The body of the robot is connected to a boom, and for this work a mechanical lock is used on the boom side to limit body rotation. An equivalent model for the system during 2D operation is shown to the right, where the SEA position L_a is a state with real dynamics and represents the physical active spring compression achieved via actuator input u_{leg} , and L represents the leg length of the robot. Since the robot is cable driven, an optical encoder on the leg pulley gives us a direct measurement of the robot's leg length. The robot is roughly 1 meter in height.

hopper 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 [10], [11], [12], and many of these methods focus on simplification of system models that have dynamics importantly different from those of a real hopping robot.

Our group is developing improved closed-form approximations for the step-to-step dynamics of planar hopping [13], toward computationally fast planning with a limited, noisy lookahead on rough terrain. Additionally, 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 strategies should potentially use elements of both feed-forward and feedback. However, implementing feedback controllers for series-elastic actuators (SEA) 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.

Our work in this paper is motivated by actual hardware, the hopping robot FRANK, shown in Fig. 1. Operation of real-world legged robotics ultimately requires leg placement on irregularly spaced footholds on potentially rough terrain [14], and for this to be achieved, accurate state tracking is

P. Terry, G. Piovan, and K. Byl are with the Robotics Laboratory, University of California at Santa Barbara, Santa Barbara, CA 93106, USA, pwterry@umail.ucsb.edu, giulia@enr.ucsb.edu, katiebyl@ece.ucsb.edu

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. Recent work has been done with monopods by Poulakakis and Grizzle [15], [16], applying hybrid zero dynamics and enforcing SLIP-based trajectories on higher order hopper models to generate stable hopping gaits. Additional SLIP-inspired work has been done for the robot ATRIAS, focused on high energy efficiency and impulse rejection [17]. Our work here in contrast focuses on considering feedback laws directly on system variables that allow us to either directly or indirectly control the trajectory of the system’s center of mass (CoM).

We have shown in previous work [18] that high-order feedback linearization based methods can be applied directly to the CoM during the stance phase, where both control inputs (torque at the leg and torque at the hip) are planned concurrently to orient the ground reaction force vector and maintain body stability. One disadvantage of such a method is not exploiting approximations and/or algorithms for the SLIP model already present in the literature. We are now working towards methods allowing us to incorporate fast analytical SLIP-based approximations in our trajectory planning. Therefore, in this work we focus on a model of FRANK omitting body angle dynamics, achieved in hardware by mechanically locking the body with respect to the leg. We leave the hip angle actuator u_{hip} completely unused (i.e., passive) during the stance phase and will investigate augmenting our control strategies to compensate for body motions in future work, as has been previously shown to be possible in [19]. We discuss hardware implementations for apex height regulation to not only present our first experimental results applying high-order partial feedback linearization (PFL) on FRANK, but also to verify that such a strategy is indeed feasible to implement. We investigate the application towards precise foothold selection in 2D simulations as to not be limited by noisy sensing of our robot, and show that by using PFL on the leg length dynamics we can use SLIP-based approximations even in the presence of complex SEA dynamics, and obtain accurate foothold placement.

The rest of the paper is organized as follows. Section II describes our 2D hopper model based on the robot FRANK, and introduces symbols and definitions used in the paper. Section III provides our PFL construction about the jounce of the leg state, while Section IV presents hardware results obtained applying PFL to the robot FRANK and regulating apex heights during forward motion. Section V illustrates how to apply these methods to incorporate SLIP-based approximations and precisely regulate step length during 2D hopping. Finally, Section VI contains conclusions and future work.

II. 2D HOPPER MODEL

In this work we consider the hopping robot FRANK, showed in Fig. 1 together with a model schematic. FRANK is tethered to a large carbon fiber boom on which all hardware is mounted. All computation is done on board the hardware

L_0	0.543 m	Natural leg length
m_s	7.59 kg	Sprung (body) mass
m_l	0.548 kg	Unsprung (leg) mass
m_e	7.11 kg	Eff. actuator mass
J_l	0.015 kg m ²	Leg inertia
K	2,389 N/m	Main Spring const.
c	0.002 m	Main Spring pre-load
k_p	245.18 N/m	Tension Spring const.
c	0.114 m	Tension Spring pre-load
K_t	0.0369 Nm/A	Motor torque const.
N	66	Motor gear ratio
γ	59.1 N/A	SEA Actuator const.
b_1	1.74 Ns/m	SEA lin. friction
f_1	0.91 N	SEA coulomb friction
b_2	1.08 Ns/m	Leg lin. friction
f_2	0.53 N	Leg coulomb friction
b_3	1.16 Ns/m	Hip lin. friction
L_B	2.12 m	Boom length
m_B	1.15 kg	Boom mass
u_{max}	15 A	Max current input

TABLE I

MODEL PARAMETERS FOR THE ROBOT FRANK

using a VersaLogic Tiger (VL-EPM-24) board, with communication between the host and target accomplished via Simulink Real Time. The body of the robot is mechanically locked at the boom-side, making the dynamics similar to a hardware realization of SLIP; however the robot model exhibits real actuator dynamics, leg inertia, and both sprung and unsprung mass. Similarly to the SLIP model, this system can be described as having a *flight phase* and a *stance phase*. During the flight phase the system follows ballistic-like dynamics based on the boom geometry, which are omitted here due to space limitations. The state X of the system during stance is given as

$$X = [\theta, L, L_a, \dot{\theta}, \dot{L}, \dot{L}_a]^T$$

where θ represents the leg angle, L represents the leg length, and L_a represents the position of the SEA. It can be shown using a Lagrangian approach that the dynamics of the stance phase are

$$\begin{bmatrix} \ddot{\theta} \\ \ddot{L} \\ \ddot{L}_a \end{bmatrix} = M^{-1} \left(C + \begin{bmatrix} -NK_t u_{hip} - b_3 \dot{\theta} \\ -b_2 \dot{L} - f_2 \text{sgn}(\dot{L}) \\ -\gamma u_{leg} - b_1 \dot{L}_a - f_1 \text{sgn}(\dot{L}_a) \end{bmatrix} \right) \quad (1)$$

where non-linear matrices M and C are both functions of X . The SEA actuator variable u_{leg} outputs current to drive L_a with motor torque constant K_t and gear ratio N . On the robot, this is accomplished using a network of rotational pulleys that connect to the leg mechanism, and sliding pulleys which drive the main springs. The parallel combination of these form an effective spring constant k , and an additional smaller spring k_p is used with a large amount of pre-load to tension the system. Non-linear frictional effects are common in SEA systems, particularly those that are cable driven, therefore we choose to include Coulomb friction terms f_1 and f_2 in addition to damping terms b_1 and b_2 . The transitions between the stance phase and the flight phase, denoted *take-off* (TO) and *touch-down* (TD), are discrete events at which additional energy in the system is lost due

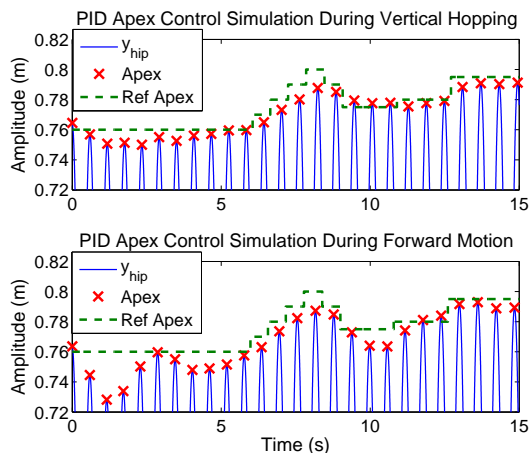


Fig. 2. This figure shows that even for the simple case of regulating apex heights (the “1D problem”) on a realistic hopper model, using simplistic control methods will not result in good performance. The first plot shows that even in vertical hopping the performance leaves much to be desired, and the second plot illustrates how performance will degrade further if such a strategy is attempted during forward motion. Such methods may be capable of converging to steady-state commands via integral action, but fast step-to-step changes require dynamical planning.

to the mass in the robot’s leg. At *touch-down*, the foot of the robot impacts the ground causing the unsprung portion of the leg’s velocity to be dissipated, and at *take-off* the moving frame of the body impacts the leg assembly causing the system to lift off the ground. The result of these impact dynamics, which are omitted here due to space limitations but can be seen in [18], are an instantaneous reduction to the velocities of the CoM at these times, with the amount lost proportional to the ratio of body (sprung) and leg (unsprung) masses. Note that while the touch-down angle can be chosen and controlled during flight, all control authority using the SEA to regulate a ballistic trajectory must be accomplished in the preceding stance phase.

Although the robot FRANK has an additional actuated angle state (θ_{\perp}), we omit it from the 2D dynamics as it is only used to prevent the leg from slipping in the perpendicular direction during operation. For this work we consider only using the SEA during nominal 2D hopping, similar to SLIP, therefore the actuator variable u_{hip} is always zero during the stance phase. Model parameters obtained via system identification experiments are shown in Table I, and are used for all hardware and simulation work presented in this paper.

III. HIGH ORDER PARTIAL FEEDBACK LINEARIZATION

This section introduces a feedback method for enforcing trajectories directly on the leg length L with realistic SEA dynamics. There are two core reasons to develop such a method. The first is simply to have an accurate method of implementing feedback control continuously on the system. Fig. 2 shows that even for the often considered “simple” case of regulating apex heights during vertical hopping, simplistic controllers that do not incorporate accurate dynamical models often have very sub-par performance, as they typically

rely on discrete energy-based estimations and/or feedback of apex state measurements only. Even in the case of analytically solving the 1D system for feed-forward commands, which is possible although not necessarily trivial, a controller solely based on feed-forward information will often result in significant errors from model inaccuracies. The second core motivation follows from the fact that analytical solutions for more general 2D hopping systems are difficult to calculate in the presence of complex actuator dynamics. In particular, by ensuring the trajectory of the leg length follows some nominal reference, approximations can be made to predict the underactuated leg angle dynamics, as these leg dynamics depend only on the leg state L and not L_a . This is explored in detail in Section V.

We would like to use u_{leg} to directly control L (not L_a), and since our system has potentially both nonlinear frictional effects and leg angle dynamics we would like to negate, we propose to use feedback linearization to accomplish both goals. Since u_{leg} does not directly affect \ddot{L} due to the presence of the series elastic spring element, we must map the feedback linearization to a higher order state than is typically used for control of underactuated mechanical systems [20], [21]. In this case, we must take two additional derivatives for our control variable to directly affect L , and therefore the relative degree of our system with L as the chosen output is 4. Since any Coulombic frictional terms present are not differentiable, we approximate them with an arctan function, which is reasonably close for scaling constant μ sufficiently small, and generate the approximate dynamics as

$$W = M^{-1} \left(C + \begin{bmatrix} -NK_t u_{hip} - b_3 \dot{\theta} \\ -b_2 \dot{L} - \frac{2f_2}{\pi i} \operatorname{atan}\left(\frac{\dot{L}}{\mu}\right) \\ -\gamma u_{leg} - b_1 \dot{L}_a - \frac{2f_1}{\pi i} \operatorname{atan}\left(\frac{\dot{L}_a}{\mu}\right) \end{bmatrix} \right) \quad (2)$$

We proceed by taking two derivatives of the state acceleration equations in (2) to obtain the dynamics for the jounce of the system as

$$\Lambda(X, u_{leg}) \approx \frac{d^2}{dt^2} W \quad (3)$$

where we have neglected the higher order terms appearing in the derivatives of u_{hip} as for this work the hip actuator is unused during stance. We lastly substitute (3) in (2) and extract the component of Λ corresponding to the leg-length state to form our control variable as

$$\ddot{\ddot{L}} = \beta_L(X) u_{leg} + \epsilon_L(X)$$

where β_L and ϵ_L are functions of the state X . Next we define our control law as

$$u_{leg} = \frac{1}{\beta_L} (-\epsilon_L + v_L). \quad (4)$$

In other words, u_{leg} 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

$$\begin{aligned} v_L &= v_1 + v_2 \\ v_1 &= K_p(L_{ref} - L) + K_d(\dot{L}_{ref} - \dot{L}) \\ v_2 &= K_{dd}(\ddot{L}_{ref} - \ddot{L}) + K_{ddd}(\dddot{L}_{ref} - \dddot{L}) \end{aligned} \quad (5)$$

which requires four total poles in the closed-loop dynamics of $L(t)$. 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:

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

Additionally, we require both references and estimates of the acceleration and jerk of the system, which can be calculated using a filter for velocity and analytical computations.

IV. APEX HEIGHT CONTROL ON HARDWARE

In this section we present hardware results implementing our PFL strategy on the robot FRANK, implementing apex height regulation during steady-state forward motion. We acknowledge this is a simpler problem than foothold placement, which we address in simulation in the next section. However, these results demonstrate that using high-order PFL on the leg state is both feasible and performs well on real hardware, that is inevitably affected by model inaccuracies and imperfect sensing, which are currently preventing us from performing more elaborate control experiments. With the ability to drive the dynamics of the leg to desired trajectories, we have direct control authority of the radial take-off velocity of the system, and can use approximations to estimate the angular dynamics, θ , as the acceleration of these states do not instantaneously depend on L_a . For the case of apex height regulation, it is sufficient to use a reduced PFL construction as

$$\begin{aligned} G_1 &= \frac{1}{m_s} (k(L_a - L + L_0 + c) - b_2 \dot{L} - m_s g) - \frac{2f_2}{\pi} \operatorname{atan}\left(\frac{\dot{L}}{\mu}\right) \\ \ddot{L} &\approx \frac{d}{dt} G_1 = -\frac{k}{m_s} (\dot{L} - \dot{L}_a) - G_1 \left(\frac{b_2}{m_s} + \frac{2f_2}{\pi \sqrt{\mu}} \frac{1}{\mu + \dot{L}^2} \right) \\ \dddot{L} &\approx \frac{d}{dt} \left(\frac{d}{dt} G_1 \right) = \tilde{\beta}_L(X) u_{leg} + \tilde{\epsilon}_L(X) \end{aligned}$$

which clearly shows the SEA coupling dynamics in that u_{leg} does not directly appear until \ddot{L}_a appears in the dynamics for the jounce of the leg. We then use the same control law from (4) with the reduced PFL coefficients $\tilde{\beta}_L$ and $\tilde{\epsilon}_L$. Utilizing this control method for the purpose of apex regulation involves designing compatible stance phase trajectories such that the terminating value of \dot{L} , i.e., the take-off velocity L_{TO} , is set corresponding to a desired apex height. The trajectories used in (5) during hardware experiments were generated online using analytical equations for the solution of a 1D hopper model, which are omitted here due to space limitations. During the stance phase u_{hip} is zero, and during the flight phase u_{hip} uses a simple controller to position the leg in a touch-down angle θ_{TD} to achieve forward hopping. For simplicity we use a Raibert-like strategy [2] to set the touch-down angle, given by

$$\theta_{TD} = \theta_0 + K(\dot{x}_r - \dot{x}_{hip}) \quad (6)$$

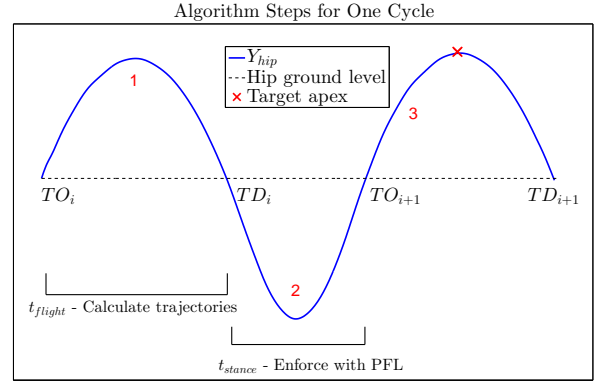


Fig. 3. To regulate an apex state, all control must be accomplished in the preceding stance phase. Furthermore, the feed-forward construction must complete all required calculations before the touch-down event, with a typical flight phase lasting 300ms-400ms, so the reference trajectories are ready at the start of the stance phase. At each step i , the algorithm begins the calculation at TO_i , forward solves impact and flight dynamics {1} to generate stance phase initial conditions TD_i , and forward solves stance and flight using a binary search to generate trajectories {2-3}. The trajectories are implemented in real time using PFL during the stance phase {2}.

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 . To obtain the correct apex heights during forward motion, we must account for the energy offset due to the non-zero leg angle at each step. Calculating this quantity entails estimating the state values at the future take-off: \dot{L}_{TO} , θ_{TO} , and $\dot{\theta}_{TO}$. However, \dot{L}_{TO} can be computed as

$$\frac{m_s}{m_l + m_s} \dot{L}_{TO} = \sqrt{2g(h_{des} - L_0)} - L_0 \dot{\theta}_{TO} \frac{m_l}{2(m_l + m_s)}$$

where h_{des} is the desired apex height, and we have used the system's impact dynamics and the fact that we are regulating $L(t)$ using PFL to force \dot{L}_{TO} , to account for energy loss at take-off. We lastly use the following simple but adequate symmetry approximation for the take-off angular variables

$$\theta_{TO} \approx -\theta_{TD}, \quad \dot{\theta}_{TO} \approx \dot{\theta}_{TD}. \quad (7)$$

These assumptions allow us to estimate the energy offset from 1D hopping that occurs during steady-state 2D motion and command a slightly higher \dot{L}_{TO} , which is the experiment performed on the hardware.

In order to adopt an apex height control strategy to the robot FRANK, several hardware details must be accounted for. Although a mechanical lock prevents the boom side of the robot from rotating, torsional forces in the carbon fiber boom result in the body of the robot flexing up to roughly 5 degrees during operation. The effect on the stance phase dynamics is largely negligible, however when controlling a specific touch-down angle θ_{TD} the relative encoder reading at the hip must be added to the body angle ϕ to correctly position the global leg angle. A pair of HC-SR04 acoustic sensors were mounted to each side of the robot in order to estimate the body angle ϕ so this could be accomplished. Since the robot hardware is tethered to a boom, during the flight phase the boom angle ψ follows non-linear dynamics that must be accounted for when determining proper take-off

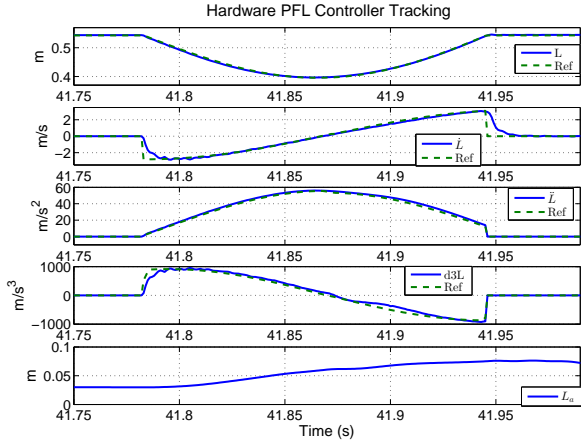


Fig. 4. The robot FRANK was successfully able to track all references during the stance phase with reasonable results. The state measurement for L is estimated using a filter, and the state measurements for \dot{L} and \ddot{L} are calculated in real-time using analytic equations. The SEA position L_a is also shown to illustrate that it remains quite well behaved.

velocities. While hopping the boom angle $\psi \in [-13^\circ, 0^\circ]$, therefore we use a small angle approximation to analytically solve and convert information from the leg states L and \dot{L} to boom flight phase states ψ and $\dot{\psi}$. Lastly, all computation is running locally on the robot hardware, which has a maximum sampling rate of 1 KHz. Determining the correct feed-forward solution requires some calculation time, and since a typical stance phase can be as short as 150 ms, all calculations must be accomplished in the preceding flight phase in an algorithmic way, as explained by Fig. 3.

Experimental tracking performance for set of trajectories is shown in Fig. 4, which verifies the feasibility of using our 4th-order PFL control on real hopper hardware. Results for apex tracking during approximate steady-state motion, with a commanded forward speed of $\dot{x}_r = 0.42 \text{ ms}^{-1}$, are shown in Fig. 5. As the results show, the robot is able to track apex references that quickly change on a step-to-step basis even in the presence of noisy forward velocity signals. This is a considerable improvement over results generated via simplistic PID methods shown in Fig. 2. We note that both accuracy of forward speed and apex height can likely be improved by more elaborate touch-down angle controllers, usage of the hip actuator in stance, and in particular incorporating model information of SLIP-like angular dynamics, which is explored in the next Section.

V. FURTHER APPLICATIONS IN 2D HOPPING

In this section we illustrate how to use the 4th-order PFL strategy to implement analytical SLIP-based trajectories and enforce step length commands. As shown in Section II, the actuator's dynamics (1) cause even controlling something as simple as the apex height to not be necessarily straightforward. The 1D case is a problem that considerably simplifies the system's dynamics by not considering the system's angular dynamics. Consequently, there exists a closed-form solution for the stance phase that allows one to design the reference trajectory for the leg-length dynamics, L_{ref} , to set

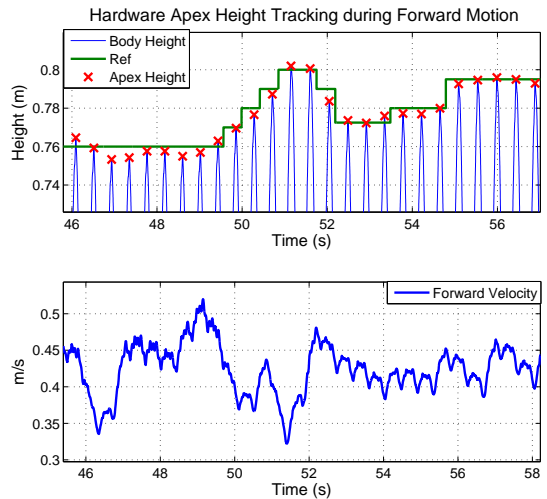


Fig. 5. The controller is able to track apex heights on the hardware, even during forward motion. By utilizing a feed-forward based algorithm, we are able to track step-to-step apex changes and also gain increased tracking performance of feedback control by using our high order PFL controller directly on the leg state. The average apex error recorded for this data is $1.6\% \pm 1.3\%$ of the commanded apex height clearance off the ground. Accuracy can likely be improved in future work by using the leg angle actuator, which is currently unused in stance, to better regulate forward speed, but comparing to Fig. 2 we clearly show hardware results that are an improvement over simulation results using conventional PID methods.

the necessary take-off velocity and drive the system to a desired apex height.

In the 2D case, regulating the ballistic phase requires controlling the take-off values \dot{L}_{TO} , θ_{TO} and $\dot{\theta}_{TO}$. For non steady-state gaits with varying step lengths, simple approximations such as (7) will not provide acceptable results as they do for the constant speed case. Since the SEA dynamics only instantaneously affect L and not θ , we can force the angular dynamics of the system to be quite similar to SLIP, for which we have fast analytical approximations, by enforcing certain trajectories on the leg using our high-order PFL controller.

A. Trajectory generation

Work in [19] considers SLIP dynamics. It is shown that if we can control exactly the leg-length dynamics to be a composite of sinewaves of the form $L(t) = r + a \cos(\omega t + \beta) + vt$, we can compute an accurate approximation of the leg-angle dynamics throughout stance as

$$\theta(t) \approx \frac{1}{L(t)} (u_0(t) + \epsilon u_1(t) + \epsilon^2 u_2(t) + \dots),$$

where the various $u_i(t)$'s for $i = 0, 1, \dots$ are solutions of

$$\begin{aligned} \frac{d^2}{dt^2} u_0 - \lambda^2 u_0 &= 0, \\ \frac{d^2}{dt^2} u_i - \lambda^2 u_i &= -\delta u_{i-1} \cos(\omega t + \beta), \quad i = 1, 2, \dots \end{aligned}$$

and the initial conditions for the above differential equations and constants r , v , λ , β , δ , ω , and ϵ are determined from the robot state at touch-down.

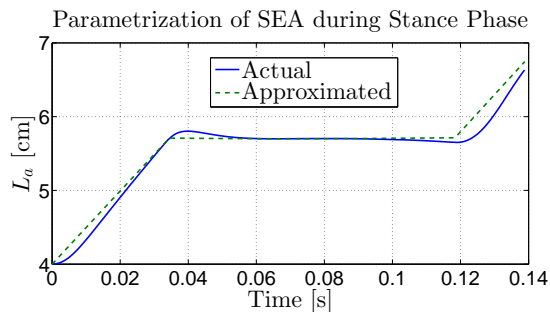


Fig. 6. This figure shows an example of the approximated $L_a(t)$ computed as in (8) (dotted green line), used to generate the leg-length reference trajectory, versus the actual L_a computed from the system's equations of motion (1) (solid blue line). The parameters used for the approximated function are $L_{a0} = 0.03$ m, and $t_s = 0.118$ s. We can see here that the nonlinear dynamics of L_a are not captured by the parametrization (8). However, as previously stated, the high-order PFL ignores the accuracy of the parametrization of L_a , and thus mismatches in the parametrization of L_a do not affect the performance of the PFL tracking, as long as it is within the current and physical limits of our system.

To solve the SLIP's leg-length dynamics to be a composite of sinewaves, [19] loosely approximates the actuator's motion to be the sum of a piecewise linear function $L_{lin}(t)$ paired with a nonlinear component, $L_{nl}(t)$, as

$$L_a(t) = L_{lin}(t) + L_{nl}(t). \quad (8)$$

$L_{nl}(t)$ is a state feedback term with the purpose of cancelling the nonlinear terms of the L -dynamics in SLIP to allow for the construction of an analytical solution, and therefore is fully determined by the states of the system throughout stance. Generating a trajectory to drive the system to a desired state becomes then a problem of finding the right piecewise linear function $L_{lin}(t)$. To do so, $L_{lin}(t)$ can be parametrized as follows. As shown in [19] and [22], to maximize the reachable space of the SLIP model, i.e., the space of all the apex states $[y, \dot{x}]$ reachable in one step, the actuator needs to be controlled throughout the entire stance phase. We assume that the actuator moves at its maximum constant velocity to reach a desired constant value, L_{a0} , and at some time t_s the actuator moves with maximum velocity towards its upper or lower limit until the end of the stance phase. Therefore, searching for an actuation policy to yield some nominal $L(t)$ is equivalent to performing a search over all feasible values of L_{a0} and t_s . It is important to note that the nonlinear dynamics of L_a are not captured by the linear parametrization that we use to generate a reference trajectory for L , as shown in Fig. 6. However, the 4-th order feedback linearization has the advantage of redirecting the problem to the dynamics of L , essentially ignoring the accuracy of the parametrization used to generate the trajectory for the leg.

B. Step-Length Control

We now provide an example application of our control strategy to generate and enforce trajectories on the leg state for 2D hopping. In particular, we control our system to hop following a reference trajectory of step-lengths, i.e., the distance between two consecutive footholds. At each take-off we use a minimization algorithm to find the values L_{a0}

and t_s that minimize the cost function

$$J = |S_i - S_{ref}|, \quad (9)$$

where S_i and S_{ref} are the achieved and desired step-lengths respectively, which are a function of both the ballistic path and angle values at take-off and touch-down.

The algorithm proceeds as follows (refer to Fig. 3).

- i At the $(i - 1)$ -th take-off, TO_{i-1} , the (i) -th touch-down angle for the following take-off TO_i is computed as in (6), with \dot{x}_r as a reference forward velocity.
- ii The reachable space for the system at step (i) , \mathbf{R}_i , is computed using the closed-form approximation of the stance-phase dynamics for a coarse grid of the parameters L_{a0} and t_s .
- iii From TO_i we can backsolve the ballistic flight phase and find the set of states that give the reference step-length S_r :

$$[y_S, \dot{x}_S] = \{[y, \dot{x}] \in \mathbf{R}_i \mid S_i = S_r\},$$

and we choose the policy

$$\{L_{a0}^S, t_s^S\} = \arg \min_{L_{a0}, t_s} |\dot{x}_S - \dot{x}_r|,$$

which represents the point in the $\{L_{a0}, t_s\}$ -grid closest to our target state.

- iv Because $\{L_{a0}^S, t_s^S\}$ is computed from a coarse grid, the optimal policy that minimizes the cost function J in (9) is found using a Nelder-Mead constraint optimization algorithm as

$$\{L_{a0}^{opt}, t_s^{opt}\} = \arg \min_{L_{a0}, t_s} J,$$

initialized at $\{L_{a0}^S, t_s^S\}$.

- v L_{a0}^{opt} and t_s^{opt} are used to compute the leg-length reference trajectory L_{ref} , and the 4-th order PFL (4)-(5) is implemented.

Thanks to the existence of an accurate closed-form approximation for the stance phase dynamics, both the reachable space computation and the optimization search can be performed with low cost in terms of computation time. In particular, generating a grid with roughly 500 points takes about a quarter of a second on a second generation i7 processor. While in general the cost function J is not convex over the entire $\{L_{a0}, t_s\}$ -space, a minimum with respect to some error tolerance will be in a neighborhood of $\{L_{a0}^S, t_s^S\}$. Additionally, $\{L_{a0}^S, t_s^S\}$ converges to $\{L_{a0}^{opt}, t_s^{opt}\}$ as the number of points in the grid used to compute the reachable space at step (ii) tends to infinity. Therefore, if the $\{L_{a0}, t_s\}$ -grid is fine enough with respect to a set tolerance for the error of the solution found, step (iv) can be omitted.

Fig. 7 shows simulation results of our control strategy for a set of desired step lengths. The reference step length is changed every 3 hops, and it is chosen from a set of feasible step lengths. We note that the usage of simplistic touch-down angle control such as (6), used only for simplicity to demonstrate our work, limits the reachable space at each step by forcing all possible solutions of a given step length to be within some neighborhood of velocity reference \dot{x}_r . We leave

the problem of optimal touch-down angle selection open for future work.

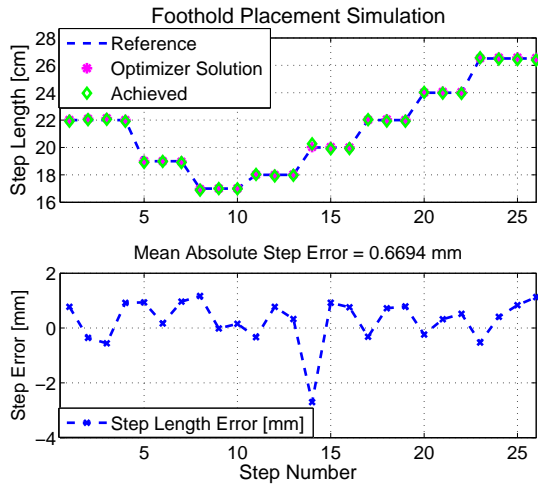


Fig. 7. Above shows simulation results using high order feedback linearization to regulate trajectories generated by the SLIP-based algorithm for the purpose of regulating step length, using parameters in Table I. The optimizer solution refers to the step length computed with the stance phase approximation, while the achieved state is computed solving numerically the system's equations of motion. The touch-down angles are set using the simple control law in (6), with $\theta_0 = -3$ deg and $\dot{x}_r = 0.4m/s$. Using such a simplistic control law for determining the touch-down angles limits the reachable space at each step, but good accuracy for feasible step commands can be achieved.

VI. CONCLUSIONS AND FUTURE WORK

We have presented motivating hardware results and algorithms for modeling and control techniques for a realistic series-elastic actuated hopping robot to achieve accurate state tracking. For practical hopping robots, actuators have real dynamics 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.

Future work will investigate both unlocking the body and using the leg angle actuator to allow comparability between body stabilization and SLIP-based trajectory generation. We seek to compare such methods to our previous work, with the ultimate goal of accurately regulating stride length, and providing a framework that allows for future work in the construction of trajectories to achieve desired results such as stride switching, operation on rough terrain, and disturbance rejection.

ACKNOWLEDGMENTS

This work is supported in part by DARPA's M3 program (Grant No. W911NF-11-1-0077) and by an NSF CAREER award (CMMI 1255018).

REFERENCES

- [1] 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.
- [2] M. H. Raibert, *Legged Robots that Balance*. Cambridge, Massachusetts: MIT Press Series in Artificial Intelligence, 1986.
- [3] R. M. Alexander, "Three uses for springs in legged locomotion," *Int. Journal of Robotics Research*, vol. 9, no. 2, pp. 53–61, 1990.
- [4] H. Geyer, A. Seyfarth, and R. Blickhan, "Compliant leg behaviour explains basic dynamics of walking and running," in *Proc. of the Royal Society*, pp. 2861 – 2867, 2006.
- [5] R. Blickhan, "The spring-mass model for running and hopping," *Journal of Biomechanics*, no. 22, pp. 1217–1227, 1989.
- [6] 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.
- [7] 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.
- [8] R. Blickhan and R. J. Full, "Similarity in multilegged locomotion: Bouncing like a monopode," *Journal of Comparative Physiology A: Neurothology, Sensory, Neural, and Behavioral Physiology*, vol. 173, pp. 509–517, Nov. 1993.
- [9] J. Seipel and P. Holmes, "A simple model for clock-actuated legged locomotion," *Regular and Chaotic Dynamics*, vol. 12, no. 5, pp. 502–520, 2007.
- [10] 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.
- [11] W. J. Schwind and D. E. Koditschek, "Approximating the stance map of a 2 DoF monopod runner," *Journal of Nonlinear Science*, vol. 10, no. 5, pp. 533–588, 2000.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] I. Poulakakis and J. W. Grizzle, "Formal embedding of the spring loaded inverted pendulum in an asymmetric hopper," in *Proc. of the European Control Conference*, pp. 3159 – 3166, July 2007.
- [16] I. Poulakakis and J. W. Grizzle, "Modeling and control of the monopodal robot thumper," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 3327 – 3334, May 2009.
- [17] D. Koepl and J. Hurst, "Impulse control for planar spring-mass running," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 3-4, pp. 589–603, 2014.
- [18] P. Terry and K. Byl, "CoM control for underactuated 2d hopping robots with series-elastic actuation via higher order partial feedback linearization," in *Proc. IEEE Conf. on Decision and Control*, 2015.
- [19] G. Piovan and K. Byl, "Approximation and control of the slip model dynamics via partial feedback linearization and two-element leg actuation strategy," *IEEE Transactions on Robotics*, 2016. In publication.
- [20] M. W. Spong, "Partial feedback linearization of underactuated mechanical systems," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 314–321, 1994.
- [21] 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.
- [22] G. Piovan and K. Byl, "Reachability-based control for the active slip model," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 270–287, 2015.