SLIP-based and HOPFL Enforced Step Length Control for Stable Body Motions of Realistic SEA Hoppers

Pat Terry, Giulia Piovan, and Katie Byl

***Abstract*— In this paper, we consider the problem of precise step length control on underactuated hopping robots, integrating analytical SLIP-based trajectories on a hardware inspired hopper model that includes series-elastic actuation (SEA), leg mass and inertia. Additionally, in the model considered the body center of mass (CoM) is not collocated with the hip joint, and must be stabilized for successful hopping. The majority of recent work with hopping robots is geared towards providing a discrete set of stable limit cycles, and is thus not well suited for non steady-state motions which are required for foothold placement in the real world. Therefore, this work focuses on developing control methods capable of dynamically switching between strides, and thus allowing footholds to be controlled on a step-to-step basis. Building on previous work, we illustrate how high-order partial feedback linearization (HOPFL) can be implemented directly on the leg-length state, and show how to apply these results so that algorithms designed for the SLIP model already widely available in the literature can be accurately implemented as reference trajectories. Furthermore, we develop control laws for the hip torque in order to both stabilize body motions and force the resulting leg angle trajectories to be analytically tractable as following the SLIP model's dynamics.**

I. Introduction

Hopping robots are relatively simple systems that still capture key features involved in dynamic locomotion. These systems are known to have many parallels towards running gaits in animals and humans [1], as well as more general and/or multi-legged robots [2]. To achieve highly compliant legs capable of dynamic running gaits with potentially high energy efficiency, we typically consider robot models with springs in series with their legs [3], [4]. This has led to the development of many hopping and other series-elastic-actuated robots [5], [6], [7], [8], including well-known work led by Raibert [9], which produced many successful but heavily tuned experimental methods and results. Despite a significant amount of literature regarding these hopping systems, precision control remains a quite challenging problem.

The prevalent model used to characterize hopping robots, and describe some general trends for running systems, is the Spring Loaded Inverted Pendulum (SLIP) model [10]. Although a quite simple model, the dynamics of this system are not analytically solvable. Many works have provided approximation techniques of varying levels of complexity [11], [12], [13], [14] in order to facilitate development of on-line state prediction and control. Most recently, [15] proposes to not only augment SLIP with a series-elastic actuator (SEA), but use this active term to simplify the leg dynamics such that approximations can be employed to a high level of accuracy. These new approximation methods result in SLIP not only being a source of inspiration when studying qualitative features of hopping/running gaits, but a tool for generating analytical hopping trajectories with low computational cost that can be used as references in more complicated hopping and/or running systems.

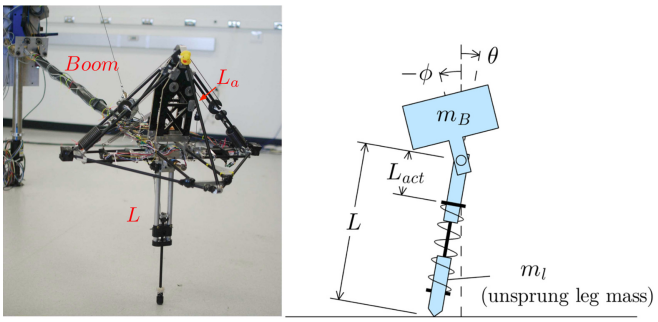

Fig. 1. The robot FRANK is a low weight SLIP-inspired underactuated hopper. The robot consists of a series elastic actuator (SEA) to inject energy into the system, a planar leg angle actuator to control the forward touch-down angle (θ), and an underactuated body angle (ϕ) which rotates during operation and must be kept upright to maintain stable motions. 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. The center of mass (CoM) of the body does not correspond to the hip joint, giving way to non-trivial body-leg coupling dynamics between ϕ and θ.

Having two actuators and four dynamic states, the hopping robot FRANK, seen in Fig. 1, is highly underactuated. Furthermore, the body CoM does not coincide with the robot's hip joint, giving rise to non-linear coupling dynamics between leg and body. The problem of stabilizing non-trivial pitch dynamics for asymmetrical hoppers has been previously studied [16], [5], [17], [18], [19], but it remains largely unsolved in terms of precise foothold placement. Fairly recent work by Poulakakis and Grizzle based on the robot Thumper [18] has shown that Hybrid Zero Dynamics can be used to embed classical SLIP solutions as steady-state trajectories for these kind of systems. While body stability is analytically provable in the case of steady-state gaits, these methods are highly complex and somewhat ill-suited for the case of foothold selection, as this can require non steady-state gaits to be feasible. Our past work in [19] provides offline trajectory generation methods that are enforced on-line directly on the CoM states, where both control inputs (torque at the leg and torque at the hip) are

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

planned concurrently to orient the ground reaction force vector and maintain body stability. However, the trajectory design process is challenging, as trajectories for desired step lengths are difficult and time consuming to construct in the presence of analytically unsolvable dynamics.

This work aims to develop methods capable of incorporating fast analytical SLIP-based approximations in our trajectory planning. In our previous work [20] we focused on a model of FRANK omitting body angle dynamics, achieved in hardware by mechanically locking the body with respect to the leg. We intentionally left the hip angle actuator u_{hip} completely unused (i.e., passive) during the stance phase, and provided motivating hardware results implementing high-order partial feedback linearization (HOPFL) on the leg state of the robot. We now consider the problem of designing SLIP-based trajectories for the leg state and augmenting them to be implemented concurrently with non-zero hip actuator control laws, which are required to provide body stability. We show that during the ballistic phase, the leg-body coupling dynamics due to the non-zero leg mass can be analytically approximated using small angle approximations. Similarly, we illustrate how to approximate the body dynamics during the stance phase, such that a control strategy can be developed to both provide body stabilization and force the leg angle dynamics to be analytically computable from our SLIP-based solutions. By pairing these method with the use of HOPFL on the leg length dynamics, we can use SLIP-based approximations even in the presence of both complex SEA and body dynamics, and obtain accurate step length results.

The rest of the paper is organized as follows. Section II overviews modelling of hopping robots, and Section III illustrates how to implement HOPLF on the robot leg sate and construct analytical SLIP-based trajectories as reference trajectories for our model. Section IV provides approximations of the leg-coupling body dynamics and develops a control law for the torque input at the hip. Lastly, Section V provides our step-length regulation algorithm and simulation results.

II. Hopper Modeling

In this section we discuss dynamical modeling of hopping robotic systems. The classical model used to describe hopping robots is SLIP, which we present to compare and contrast with the more complex model of our robot FRANK.

A. The Active SLIP Model

The trajectories generated for the leg state in this work are partially generated by approximate solutions to the active SLIP model. The SLIP model is a hybrid system, and its dynamics can be divided in two phases, as in Fig. 2: a flight phase, purely ballistic with respect to the center of mass; and a stance phase, characterized by ground contact between the foot and the ground. In the active SLIP, the leg is equipped with a series-elastic actuator to artificially compress/extend the spring, thus varying the system's net energy. The equations of motion for the stance phase can be

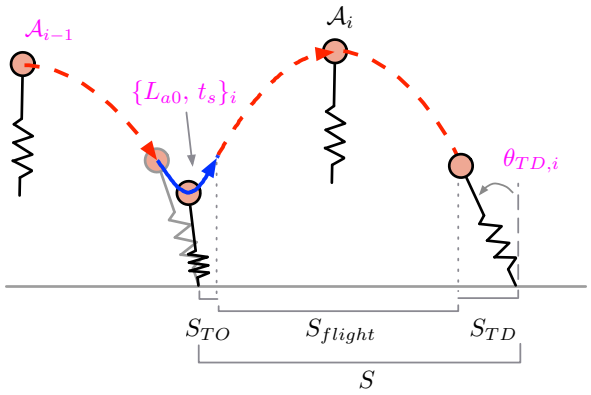

Fig. 2. SLIP is a classical model often used to describe hopping gaits. The Active SLIP model includes an active input term $L_a(t)$ to allow the spring compression to be changed. The typical control variable of interest is the step length S, which is the sum of three components based on the take-off/touch-down angles and the ballistic trajectory. Building on previous work, we can analytically compute approximations for the apex state $\mathcal{A} = [y,\, \dot{x}]$ by parametrizing $L_a(t)$ with two variables L_{a0} (set value) and t_s (switch time). By calculating $\mathcal{A}_i$ and the next touch-down angle $\theta_{TD,i+1}$, the step length S can be analytically approximated to very high accuracy.

easily computed via a Lagrangian approach, and expressed in polar coordinates as:

$$\ddot{L} = -\frac{k}{m}(L - L_0 - L_a) - g\cos\theta + L\dot{\theta}^2 \tag{1}$$

$$\ddot{\theta} = 2\frac{\dot{L}\dot{\theta}}{L} - \frac{g}{L}\sin\theta \tag{2}$$

where L is the leg length and θ is the angle the leg forms with respect to the ground, as shown in Fig. 2. Additionally, k is the spring stiffness, m is the mass of the body, and L_a is the actuator's length.

B. 2D SEA Hopper Model

The model we study for this work is based on the robot FRANK (FRANK: Robot Acronym Not Known), seen in Fig. 1. Note that our angle and state conventions largely parallel Raiberts original work [16]. The robot has both a body mass non collocated with the hip joint and non-zero leg mass and inertia. Similar to SLIP, this system can be described as having a flight phase and a stance phase. During the flight phase the system follows ballistic dynamics (omitted here due to space limitations). During the stance phase the states X of the system are given as

$$X = [\theta, \phi, L, L_a, \dot{\theta}, \dot{\phi}, \dot{L}, \dot{L}_a]^T \tag{3}$$

where θ represents the leg angle, ϕ represents the body 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

$$\begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \\ \ddot{L} \\ \ddot{L}_a \end{bmatrix} = M^{-1}\left(C + \begin{bmatrix} -NK_t u_{hip} \\ NK_t u_{hip} \\ -b_2\dot{L} - f_2 sgn(\dot{L}) \\ -\nu u_{leg} - b_1\dot{L}_a - f_1 sgn(\dot{L}_a) \end{bmatrix}\right) \tag{4}$$

m_B	7.59 kg	Sprung (body) mass
m_l	0.548 kg	Unsprung (leg) mass
m_e	7.11 kg	Effective actuator mass
J_l	0.015 $kg\ m^2$	Leg inertia
J_B	0.227 $kg\ m^2$	Body inertia
l_0	54.3 cm	Natural leg length
l_1	16.2 cm	Body CoM dist. from hip
K	2,389 N/m	Main spring constant
c	0.002 m	Main spring pre-load
k_p	245.18 N/m	Tension spring constant
c	0.114 m	Tension spring pre-load
K_t	0.0369 Nm/A	Motor torque constant
N	66	Motor gear ratio
ν	59.1 N/A	SEA actuator constant
b_1	1.74 Ns/m	SEA linear friction
f_1	0.91 N	SEA coulomb friction
b_2	1.08 Ns/m	Leg linear friction
f_2	0.53 N	Leg coulomb friction
u_{max}	20 A	Maximum current input

TABLE I

MODEL PARAMETERS FOR THE ROBOT FRANK

where matrices M and C are both functions of X. The actuator variable u_{hip} outputs current to drive the leg angle actuator on the robot with motor torque constant K_t and gear ratio N. Similarly, u_{leg} outputs current to drive the SEA via motor force constant ν. The coefficients b and f represent damping and Coulomb frictional constants respectively. In the SLIP model $L_a(t)$ is the control variable and can typically be chosen freely: the input/output pair is $(L_a,\, L)$. For more realistic implementations such as FRANK, however, $L_a(t)$ itself is typically driven by a motor torque, τ (or, equivalently, an input current u_{leg}), thus the SEA position $L_a(t)$ is an additional degree of freedom and cannot change instantaneously as in the Active SLIP model.

The transitions between stance and flight, denoted *take-off* (TO) and *touch-down* (TD), are discrete events at which additional energy in the system is lost due to the mass in the robot's leg. The result of these impact dynamics, which are omitted here due to space limitations but can be seen in [19], is 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. All simulation results presented in this paper use realistic model parameters obtained via system identification of the robot FRANK, and can be seen in Table I.

III. HOPFL CONTROL OF SLIP-BASED TRAJECTORIES

This section illustrates how to build the HOPFL controller on the leg length state, implementing analytical SLIP-based solutions as reference trajectories. An expanded discussion along with motivating hardware results on FRANK with the body mechanically locked can be seen in [20]. Construction of the body stabilizing hip torque controller and the complete step length regulation algorithm are then provided in Sections IV and V respectively.

A. HOPFL Construction

We want 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 [21], [22]. In this case, we must take two additional derivatives for our control variable u_{leg} to directly affect L. We approximate the non-differentiable Coulombic terms with a scaled $\arctan$ function, and generate the approximate dynamics as

$$W = M^{-1}\left(C + \begin{bmatrix} -NK_t u_{hip} \\ NK_t u_{hip} \\ -b_2\dot{L} - \frac{2f_2}{pi} atan(\frac{\dot{L}}{\mu}) \\ -\nu u_{leg} - b_1\dot{L}_a - \frac{2f_1}{pi} atan(\frac{\dot{L}_a}{\mu}) \end{bmatrix}\right) \quad (5)$$

We proceed by taking two derivatives of the state acceleration equations in (5) to obtain the dynamics for the jounce of the system as

$$\Lambda(X, u_{leg}, u_{hip}, \dot{u}_{hip}, \ddot{u}_{hip}) = \frac{d^2}{dt^2} W \quad (6)$$

Note that this differs from our body-locked work in [20], where u_{hip} was left unused. We lastly substitute (6) in (5) and extract the component of Λ corresponding to the leg-length state to form our control variable as

$$\dddot{\dddot{L}} = \gamma_L u_{leg} + \beta_L u_{hip} + \eta_L \dot{u}_{hip} + \alpha_L \ddot{u}_{hip} + \epsilon_L \quad (7)$$

where all coefficients are functions of the state X. In order to define the control law for L we must calculate u_{hip}, $\dot{u}_{hip}$, and $\ddot{u}_{hip}$, which will be provided in Section IV when we account for body motions. When we take two derivatives of u_{hip}, however, state feedback terms will typically cause the term $\ddot{L}_{act}$ to appear, and therefore the component of $\ddot{u}_{hip}$ due to u_{leg} must be separated and used to generate new effective coefficients as

$$\begin{aligned} \ddot{u}_{hip} &= \delta_1 + \delta_2 u_{leg} \\ \tilde{\gamma}_L &= \gamma_L + \alpha_L \delta_2 \end{aligned} \quad (8)$$

Finally, using equations (7) and (8) we define our control law as

$$u_{leg} = \frac{1}{\tilde{\gamma}_L}(-\epsilon_L - \beta_L u_{hip} - \eta_L \dot{u}_{hip} - \alpha_L \delta_1 + v_L) \quad (9)$$

In other words, u_{leg} cancels the leg's natural dynamics and forces any errors in L to decay via linear, fourth-order dynamics that are set through v_L. Specifically, we choose

$$\begin{aligned} v_L &= v_1 + v_2 \\ v_1 &= K_p(L_{ref} - L) + K_d(\dot{L}_{ref} - \dot{L}) \\ v_2 &= K_{dd}(\ddot{L}_{ref} - \ddot{L}) + K_{ddd}(\dddot{L}_{ref} - \dddot{L}) \end{aligned} \quad (10)$$

To set the controller gains we first select a dominant pole-pair with natural frequency ω_n and damping ratio ζ, and then set a 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:

$$\begin{aligned} K_p &= p_3 p_4 w_n^2 \\ K_d &= (p_3 + p_4) w_n^2 + 2 p_3 p_4 \zeta w_n \\ K_{dd} &= w_n^2 + 2\zeta(p_3 + p_4) w_n + p_3 p_4 \\ K_{ddd} &= p_3 + p_4 + 2\zeta w_n \end{aligned} \tag{11}$$

Additionally, we require both references and estimates of the acceleration and jerk of the system, which can be calculated using analytical computations once u_{hip} has been defined.

B. Leg State Trajectory Generation

The SEA dynamics only instantaneously affect L and not θ. As was shown in our previous work [20], we can force the angular dynamics of the system to be quite similar to SLIP, for which fast analytical approximations exist, by enforcing desired trajectories on the leg using our HOPFL controller. Work on SLIP in [15] shows that by controlling the leg dynamics to be 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

$$\theta_{SLIP}(t) \approx \frac{1}{L(t)}(u_0(t) + \epsilon u_1(t) + \epsilon^2 u_2(t) + ...), \tag{12}$$

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

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

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. To solve the SLIP's leg-length dynamics to be a composite of sinewaves, [15] 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

$$L_a(t) = L_{lin}(t) + L_{nl}(t) \tag{13}$$

$L_{nl}(t)$ 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. Consequently, controlling the system requires parametrizing $L_{lin}(t)$, which can be accomplished as follows. As shown in [15] and [23], to maximize the reachable space of the Active SLIP model, i.e., the space of all the apex states $\mathcal{A}$ reachable in one step, the actuator must move 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. Since this approximation is completely analytic, these calculations are extremely fast and can be implemented on-line. Therefore, given a desired step length and initial conditions, typically at *take-off*, we can solve the ballistic dynamics to determine stance phase initial conditions, search for the proper L_{a0} and t_s, and use the resulting $L(t)$ as references for (10) in stance.

Although the parametrization of L_a does not capture the actual dynamics the SEA will follow, we are only using this parametrization to generate trajectories for $L(t)$, which will be enforced using our HOPFL controller. Thus, as long as the SEA remains within physical limits, these discrepancies do not cause errors in our step length control. It is also of note to mention that since the SLIP model does not have any notion of unsprung mass, impact dynamics must be analytically computed for each grid point and used incorporate the appropriate amount of energy loss of each point accordingly.

IV. Body Angle Compensation

In this section, we examine how to apply our SLIP-based work to the underactuated 2D hopper model of FRANK seen in Fig. 1. As was shown originally by Raibert in [16], the body dynamics can be condensely expressed as

$$J\ddot{\phi} = F_t(t) l_1 sin(\phi - \theta) - F_n(t) l_1 cos(\phi - \theta) + \tau_{hip}, \tag{14}$$

where l_1 is the body CoM displacement from the hip. $F_t(t)$ and $F_n(t)$ are forces acting at the hip between the leg and body, and are generally non-linear functions of the states of the robot. Here we clearly see that when the body CoM is located at the hip (i.e., $l_1 = 0$), the dynamics become trivial, as was the case in [15]. In order to develop suitable control methods for the more complex dynamics considered in this work, we must first employ some approximations.

A. Approximation of Body Angle Dynamics: Flight

During the flight phase, all spring forces are zero, simplifying the dynamics considerably. However, due to the non-zero leg mass and inertia, when the leg moves during flight to be positioned in some nominal touch-down angle, its motion will inevitably impart torques on the body. It was shown in [19] that if we assume that the leg angle is controlled to follow some nominal trajectory $\theta_{fl}(t)$, we can write the flight phase body dynamics as

$$\ddot{\phi}_{fl} = f_{fl}(\phi, \dot{\phi}, \theta_{fl}, \dot{\theta}_{fl}, \ddot{\theta}_{fl}) \tag{15}$$

We next employ small angle approximations on all trigonometric terms, i.e., θ, ϕ, $(\phi - \theta)$, and $(2\phi - 2\theta)$. To force the dynamics to be analytic given θ_{fl}, we remove the three high-order terms containing ϕ and $\dot{\phi}$, resulting in

$$\begin{aligned} \ddot{\phi}_{fl} &\approx \frac{c_1 \ddot{\theta}_{fl} - 2 l_1 m_B m_l l_0 \dot{\theta}_{fl}^2 \theta_{fl})}{c_2} \\ c_1 &= -4 J_l m_B + 4 J_l m_l + m_B m_l l_0^2 + 2 l_1 m_B m_l l_0 \\ c_2 &= 4 m_B m_l l_1^2 + 2 m_B m_l l_0 l_1 + 4 J m_B + 4 J m_l \end{aligned} \tag{16}$$

where the terms omitted are $2 l_1 m_B m_l l_0 \theta_{fl} \dot{\phi}^2$, $2 l_0 l_1 m_B m_l \dot{\theta}_{fl}^2 \phi$, and $2 l_0 l_1 m_B m_l \dot{\phi}^2 \phi$, and are generally

expected to be quite small. Thus, we need only define a smooth trajectory for θ_{fl} and its derivatives, and then given body initial conditions we can use (16) to compute the resulting body trajectory during flight, which can be carried out with low computational cost. All flight phase trajectories $\theta_{fl}(t)$ used in this work are simply constructed to be smooth curves that terminate with a desired touch-down angle and zero angular velocity by the end of the flight phase.

B. Approximation of Body Angle Dynamics: Stance

The stance phase body dynamics are considerably more complex due to the time-varying leg length. However, since we have developed a control law that acts directly on the leg state, we have the advantage of knowing the leg length trajectory a priori. If we choose the leg-length reference trajectory for our HOPFL in (9) to be a nominal SLIP-based trajectory, we have $L(t) = L_{SLIP}(t)$, which is fully available to us. The dynamics for the leg and body angle during the stance phase can then be approximated as

$$\begin{aligned}\ddot{\theta} &\approx 2\frac{\dot{L}\dot{\theta}}{L} - \frac{g}{L}\sin\theta + \Omega_\theta(L)u_{hip} + \xi(L)(\phi - \theta) \\ &= \ddot{\theta}_{SLIP} + \Omega_\theta(L)u_{hip} + \xi(L)(\phi - \theta)\end{aligned} \tag{17}$$

$$\ddot{\phi} \approx \Omega_\phi(L)u_{hip} - \frac{L}{l_1}\xi(L)(\phi - \theta) \tag{18}$$

where Ω_ϕ, Ω_θ, ξ are non-linear differentiable functions of L and we immediately see part of the approximated leg angle dynamics are in fact those of SLIP from (2). The approximation procedure to derive the equations above parallels the approximation carried out in the previous Section, where we used small angle approximations and removed higher order terms; in this case, the specific terms removed are omitted here due to space limitations.

The accuracy of our proposed approximations for both stance and flight phase are subject to the validity of the small angle approximation and the removal of higher order terms assumption. In general, we do not expect our SLIP-inspired leg angle trajectories to be large during operation, and our control strategies will aim to keep ϕ small; hence, we expect our approximations to perform quite well in the range of operation of FRANK, albeit with some small error. Fig. 3 shows an example of our approximations for both stance and flight phase, where it is clear they perform quite well for the angles our systems operates at.

C. PFL Law for Hip Torque

In the similar though simplified case investigated in [15], a constant torque input was implemented to provide body stabilization, allowing all SLIP-based trajectories to be modified with respect to only one variable parameter that needed be numerically found. Although we cannot simply command a constant torque in this case, we can force the mapping from $\theta_{SLIP}(t)$ to $\theta(t)$ to also be a function of only one variable. We proceed by defining our control law for the hip input as

$$u_{hip} = \frac{1}{\Omega_\theta}(-\xi(\phi - \theta) + \tilde{u}) \tag{19}$$

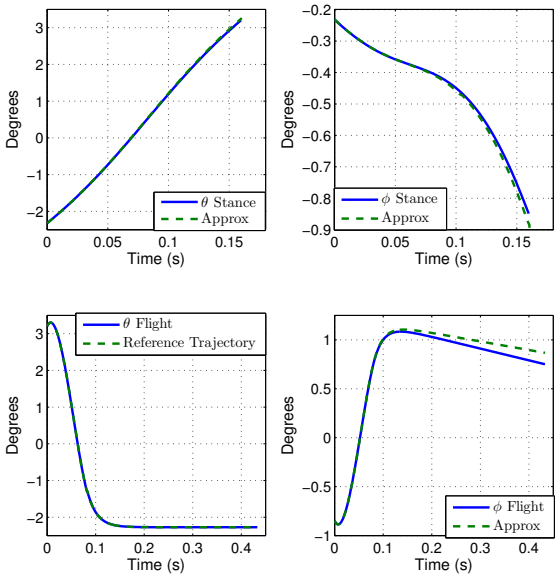

Fig. 3. Above shows our analytical approximations (green dashed line) for the leg-body coupling dynamics plotted against full dynamical simulations (solid blue line), with stance phase approximations are shown on the top, and the flight phase approximations at the bottom. In this particular example, the robot is moving forward at approximately $0.5m/s$. The reference trajectory for the leg angle in flight is simply a smooth curve terminating at a desired touch-down angle with zero angular velocity by the end of flight.

Where $\tilde{u}$ is a constant input term. The purpose of this control law is to force the approximate dynamics of the leg angle in (17) to converge to $\ddot{\theta} = \ddot{\theta}_{SLIP} + \tilde{u}$. Since we also have $L(t) = L_{SLIP}(t)$, we compute

$$\theta(t) = \theta_{SLIP}(t) + \frac{1}{2}\tilde{u}t^2 \tag{20}$$

where $\theta_{SLIP}(t)$ is calculated using our approximation in (12). Since all terms in (19) are functions of only state variables and constants, we can easily define the first two derivatives and implement the control law (19) with our HOPFL in (8)-(9).

Next, we consider the resulting body dynamics by combining (19), (20), and (18) resulting in

$$\ddot{\phi} = \frac{\Omega_\phi}{\Omega_\theta}\tilde{u} - \xi(\frac{L}{l_1} + \frac{\Omega_\phi}{\Omega_\theta})(\phi - \theta_{SLIP}(t) - \frac{1}{2}\tilde{u}t^2) \tag{21}$$

Since we can generate solutions for $L_{SLIP}(t)$ and $\theta_{SLIP}(t)$, the only term preventing us from integrating (21) to generate our solution is the term ϕ itself. Therefore, we perform the following. We first evaluate (21) with the assumption $\phi(t) = \phi_0$, perform two integrations using stance phase initial conditions and generate a first order guess $\tilde{\phi}(t)$. Then, we recursively repeat the process by evaluating (21) with $\phi(t) = \tilde{\phi}(t)$. Fig 4 illustrates that this method is quite accurate after only one additional iteration, thus given $L_{SLIP}(t)$, $\theta_{SLIP}(t)$, and $\tilde{u}$ we can approximate $\phi(t)$ with two total evaluations of (21) along with four total integrations.

The purpose of the term $\tilde{u}$ is to provide a constant input variable that can be algorithmically searched over to find

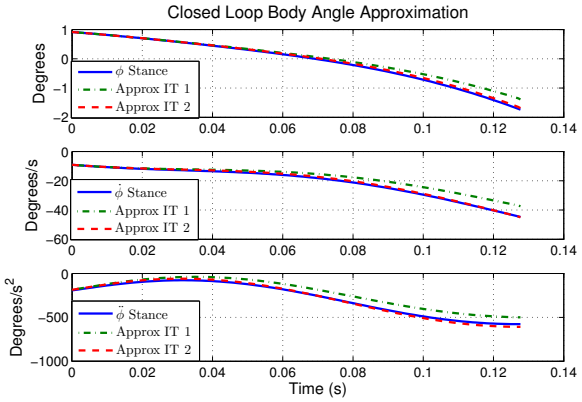

Fig. 4. Using the control law in (19) allows us to approximate the stance body angle dynamics using only initial conditions, a corresponding SLIP trajectory, and $\tilde{u}$. The actual trajectories for the body (solid blue curve) are generated by using an ODE solver to simulate the full stance dynamics. In this example, the approximations shown are generated by using (20) to analytically compute $\theta(t)$ and integrate (21), which requires only two total iterations to achieve good accuracy, and takes less than 1ms to compute each trajectory.

solutions that provide body stabilization, similar to [15]. As was discussed in Section III, we need only search over L_{a0}, t_s and now also $\tilde{u}$ to find solutions that both result in desired step lengths and keep the body angle well behaved, which is discussed in the next section.

V. Step Length Control

For each set of initial conditions and input parameters L_{a0}, t_s, and $\tilde{u}$, we analytically compute estimates of the stance and flight phases of the system. Similar to the example provided in our previous work in [20], we control our system to hop following a reference trajectory of step-lengths . For simplicity we use a Raibert-like strategy [16] to set the touch-down angle, given by

$$\theta_{TD} = \theta_0 + K(\dot{x}_r - \dot{x}_{hip}) \tag{22}$$

where θ_0 is a constant touch-down angle, and $\dot{x}_{hip}$ is the forward velocity of the hip joint. At each take-off we use a minimization algorithm to find the values L_{a0}, t_s, and $\tilde{u}$ that minimize the cost function

$$J = |S_i - S_{ref}|, \tag{23}$$

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 steps i, ii, iv, v, and vi parallel steps i, ii, iii, iv, and v respectively from our previous work in [20], with the additional search parameter $\tilde{u}$ and body minimization step iii added. The algorithm proceeds as follows (refer to Fig. 2).

i From initial conditions, the $(i+1)-$th touch-down angle for the following take-off TO_i is computed as in (22). This allows computation during flight.

ii The reachable space for the system at step (i), $\mathbf{R}_i$, is computed using the closed-form SLIP-based approximations from Section III along with body angle approximations from Section IV for a coarse grid of the parameters L_{a0}, t_s, and $\tilde{u}$.

iii From TO_i we can backsolve the flight phase using ballistic dynamics and (16). We remove all points from $\mathbf{R}_i$ that result in the next touch-down body angle being larger than a threshold to generate $\hat{\mathbf{R}}_i$:

$$\hat{\mathbf{R}}_i = \{\mathbf{R}_i \;\mid\; |\phi_{TD,i}| < \phi_{thresh}\},$$

iv We next find the set of states that give the reference step-length S_r:

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

For each step length, there exists a curve of possible solutions. We choose the policy

$$\{L_{a0}^S, t_s^S, \tilde{u}^S\} = \underset{L_{a0},\, t_s,\, \tilde{u}}{\arg\min} |y_S - y_r|,$$

which represents the point in the $\{L_{a0}, t_s, \tilde{u}\}$-grid closest to some target state y_r, which gives us some control of the overall energy level at apex.

v Because $\{L_{a0}^S, t_s^S, \tilde{u}^S\}$ is computed from a coarse grid, the optimal policy that minimizes the cost function J in (23) is found using a Nelder-Mead constraint optimization algorithm as

$$\{L_{a0}^{opt}, t_s^{opt}, \tilde{u}^{opt}\} = \underset{L_{a0},\, t_s,\, \tilde{u}}{\arg\min}\, J,$$

initialized at $\{L_{a0}^S, t_s^S, \tilde{u}^S\}$ with constraint $|\phi_{TD,i}| < \phi_{thresh}$.

vi L_{a0}^{opt}, t_s^{opt}, and $\tilde{u}^{opt}$ are used to compute the leg-length reference trajectory L_{ref}, and the HOPFL in (9) along with the hip controller in (19) is implemented at touch-down.

Similar to our work in [20], this algorithm can be implemented on-line. While in general the cost function J is not convex over the entire $\{L_{a0}, t_s, \tilde{u}\}$-space, we need only keep ϕ within some small bounds and the global minimum for a given step length will be in a neighbourhood of $\{L_{a0}^S, t_s^S, \tilde{u}^S\}$. The number of points used in the grid to compute step (ii) is typically a function of computational time requirements. If the grid resolution is high enough, it is generally possible to omit step v and still retain acceptable performance.

Fig. 5 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. These results do exhibit some error due to our approximations, slightly higher than the body-locked case in our previous work [20], but the foothold errors are still quite reasonable. As shown, the body angle exhibits non steady-state behavior but is well behaved as any errors do to our approximations are essentially reset at each step when a new set of initial conditions are measured.

Using classical methods for touch-down angle control such as (22), 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 the velocity reference. We leave the problem of optimal touch-down angle selection open for future work.

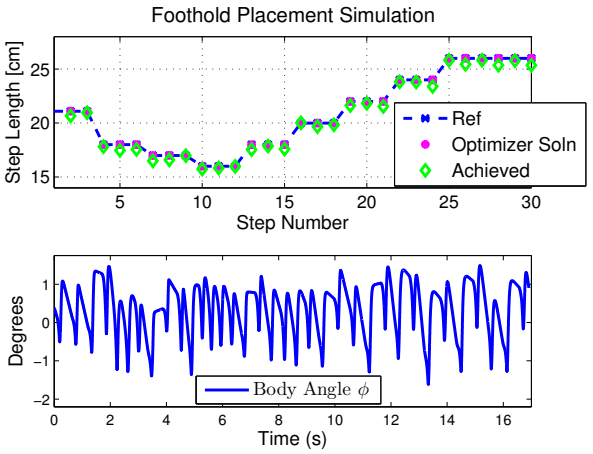

Fig. 5. Above shows simulation results during step length regulation. The optimizer solution refers to the step length computed with the stance phase approximation. The touch-down angles are set using the simple control law in (22), with $\theta_0 = -3\,\text{deg}$ and $\dot{x}_r = 0.4m/s$. The step length algorithm parameters ϕ_{thresh} and y_r used in this example are $-1.5\,\text{deg}$ and $0.85m$ respectively. The mean error of the step length results (top figure) is 1.4%, with the maximum error being only 3.5%. The resulting trajectory of the body during the simulation is also shown (bottom figure) to illustrate that while it does not exhibit steady-state behaviour it remains both well behaved and small in magnitude, by design.

VI. Conclusions

In this work we have presented 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.

We have shown that we can use HOPFL directly on the leg state of the robot, allowing us to make use of fast and analytical SLIP-based reference generation. By constructing a PFL controller for the hip torque input, we are able to provide stabilizing body motions as well as retaining the ability to analytically compute our trajectories from SLIP. The ability to perform analytical reachability computations at each step is powerful: it allows us to construct control frameworks that not only result in excellent performance but also allow for a large set of possible footholds at each step. We argue this is critical in improving both reliability (e.g., ability to recover from terrain perturbations) as well as agility (e.g., ability to accurately go to any of a family of reachable future states) of realistic spring-legged robots.

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] 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.

[2] 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.

[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] S. H. Hyon and T. Mita, "Development of a biologically inspired hopping robot-"kenken"," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 4, pp. 3984–3991 vol.4, 2002.

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

[7] G. Zeglin and B. Brown, "Control of a bow leg hopping robot," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 1, pp. 793–798 vol.1, May 1998.

[8] 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.

[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] R. Blickhan, "The spring-mass model for running and hopping," *Journal of Biomechanics*, no. 22, pp. 1217–1227, 1989.

[11] 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.

[12] 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.

[13] 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.

[14] M. L. H. Yu and H. Cai, "Approximating the stance map of the slip runner based on perturbation approach," in *IEEE Int. Conf. on Robotics and Automation*, 2012.

[15] 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*, in publication.

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

[17] S. Hyon and T. Emura, "Symmetric walking control: Invariance and global stability," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1443–1450, April 2005.

[18] I. Poulakakis and J. W. Grizzle, "The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper," *IEEE Transactions on Automatic Control*, vol. 54, pp. 1779–1793, Aug 2009.

[19] 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.

[20] P. Terry, G. Piovan, and K. Byl, "Towards precise control of hoppers: Using high order partial feedback linearization to control the hopping robot FRANK," in *Proc. IEEE Conf. on Decision and Control*, 2016.

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

[22] 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.

[23] 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.