

# Two-Element Control for the Active SLIP Model

Giulia Piovan and Katie Byl

**Abstract**—The Spring Loaded Inverted Pendulum has been extensively studied and used as an inspiration to the study of legged locomotion. Biological data suggest that legs regulate energy production and removal via muscle activation, and therefore the conservative SLIP model cannot fully explain the robustness of many legged animals during running and hopping gaits. In this work we consider the active SLIP model, an energetically non-conservative version of the SLIP model with series actuation. In particular, we propose a strategy for actuator displacement to add/remove energy from the system, and to analytically solve part of its dynamics. Additionally, we develop a control strategy for online actuator displacement to drive the system to a desired state, even in the presence of terrain perturbation.

## I. INTRODUCTION

Consisting of a mass mounted on a massless spring leg, the Spring Loaded Inverted Pendulum (SLIP) is commonly used as a simple yet effective way to model running and hopping gaits for a variety of animals. The classic SLIP model is energetically conservative. This can pose a limitation on the study of locomotion on uneven terrain, since the ability of producing or removing potential energy is essential when terrain varies in height. Furthermore, the lack of actuation precludes the possibility of correcting the system trajectory during stance when the system is subject to noise (e.g., sensor noise on terrain characteristics or on the states of the systems). This work considers the active SLIP, i.e., an actuated version of the SLIP model that allows energy variations through compression and decompression of the spring using a series actuator. As a consequence of the underactuated nature of the model, developing control laws for actuator movement is not a trivial task. Here, we propose a control action for actuator displacement in order to drive the system to a desired state. Furthermore, we exploit the benefits of the added actuator by introducing a strategy to improve existing approximations to the equations of motion of the system's dynamics.

Since its introduction in [1], the classic SLIP has been the main tool to model running behavior for a variety of animals in biomechanical studies (e.g. [2] and [3]), as well as robotics. Perhaps one of the first and most famous SLIP-based one-legged robots is the Raibert hopper [4], but several other robots have also been developed based on the one-legged bouncing model (e.g. [5] and [6]). Consequently, various control strategies have been proposed. Some studies focus on negotiating uneven terrains: e.g., [7] propose a control action to keep the running speed constant, while [8] propose an algorithm for trajectory planning, robust to model uncertainty and measurement noise. One of the main limitation of the SLIP model is the lack of a closed-form solution

for its stance phase dynamics. This problem has been the object of extensive study, and analytical approximations to the nonlinear dynamics have been provided by several researchers (e.g., [9], [10], [11], [12]). These approximations all work well in case of symmetric motion, and some include corrections to consider the effect of gravity when the motion is non-symmetric. However, as the stance phase moves away from the non-symmetric case, the approximation error becomes not neglectable, with the consequent need of an improved approximation.

Another interesting issue arises from biological data: studies on running insects [13] and guinea fowls [14] suggest that legs regulate energy production and removal via actuation, and therefore the conservative SLIP model cannot fully explain the robustness of many legged animals during running and hopping gaits. Hence, the need to examine the effect of adding actuators to the system to manage energy variations. In [15], the authors modify the SLIP with a clock based torque at the hip, and then study its stability properties. More recently, the consequences and possible applications of adding a leg actuator in series with the spring has been investigated in [16], [17], and [18]. In [16], the authors consider energy variation during stance to produce asymptotically stable gaits, while [17] focus on a control action for actuator displacement that allows for an analytic solution to the stance phase equations.

This work adds several contributions. First, we consider the active SLIP model and propose a two-part strategy for actuator displacement. On one hand, it allows us to analytically solve the equation that describes the leg-length dynamics during stance, and on the other hand it allows us to add or remove energy from the system. We then provide a feasibility study of our strategy for different system parameters. Second, we present an approximation of the leg-angle dynamics during stance when the actuator moves with constant velocity. Third, we develop a control strategy for online actuator displacement, that drives the system to a desired apex state. Finally, we extend our strategy to deal with terrain perturbations, and, for a set of system parameters, we quantitatively define the number of steps necessary for full recovery.

The paper is organized as follows. Section II reviews the setup and dynamics of the passive and active SLIP models, and introduces some symbols and definitions that will be used throughout the paper. Section III presents a novel strategy for actuator displacement. Section IV explores the applicability of the proposed strategy and provides a study of its performance. Section V develops a control strategy to reach a desired apex state, and to reduce bias on error. Section VI shows the performance of the proposed control actions with perturbation on terrain. Finally, Section VII contains conclusions and future work, and the Appendix provides our calculations for the approximation of the leg

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

angle dynamics during stance for the active SLIP.

## II. PASSIVE AND ACTIVE SLIP MODEL

In this section we review the structure of the passive SLIP model and its dynamics, and we introduce its actuated version, the so-called active SLIP model.

### A. System Dynamics

The passive SLIP is modelled as a point mass,  $M$ , attached to a massless spring leg, with length  $\ell$  and spring stiffness constant  $k$ . Running dynamics for the SLIP model consist of two phases (see Fig. 2(a)): the *flight phase*, where the body is in the air and follows a ballistic trajectory; and the *stance phase*, where the terminal part of the leg is in contact with the ground, and the compression/decompression of the spring completely defines the mass dynamics. As shown in Fig. 1(a), we define  $\ell(t)$  as the leg-length as a function of time, and  $\theta(t)$  as the leg-angle measured counterclockwise with respect to the positive horizontal axis, while  $\ell_k$  is the spring length, and  $\ell_{k,0}$  is the spring length at equilibrium. As customary,  $g$  is the gravitational acceleration. Then, the equations of motion in polar coordinates for the stance phase can be written as follows:

$$\ddot{\ell} = -\frac{k}{M}(\ell_k - \ell_{k,0}) - g \sin \theta + \ell \dot{\theta}^2, \quad (1)$$

$$\ddot{\theta} = -2\frac{\dot{\ell}}{\ell}\dot{\theta} - \frac{g}{\ell} \cos \theta. \quad (2)$$

Note that, despite their simplicity, equations (1)-(2) are not analytically solvable.

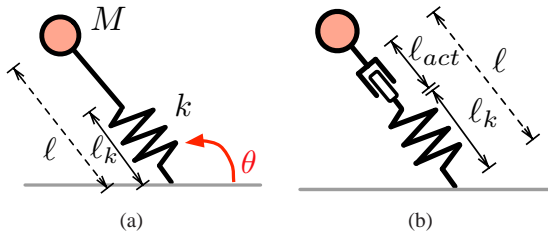


Fig. 1. 1(a) illustrates the classic SLIP model (passive SLIP), and 1(b) shows the active SLIP model (active SLIP)

The state of the mass can be easily converted into Cartesian coordinates  $x(t) = \ell(t) \cos \theta(t)$ ,  $y(t) = \ell(t) \sin \theta(t)$ , and their time derivatives  $\dot{x}(t)$  and  $\dot{y}(t)$ . We will call *touch-down (TD)* the instant that marks the transition from flight to stance; and *take-off (TO)* the instant that marks the transition between stance and flight. The highest point reached by the mass during flight is called the *apex state*, and it is defined by zero vertical velocity, i.e.,  $\dot{y}_a = 0$  [m/s]. Therefore, the apex state is completely defined by a three-dimensional vector  $\mathbf{s} = \{x_a, y_a, \dot{x}_a\}$ . In this paper we will focus our attention on the height and velocity at apex only, disregarding the forward position. Therefore, the apex state becomes a two-dimensional vector  $\mathbf{s} = \{y_a, \dot{x}_a\}$ .

The classic SLIP model is energetically conservative. However, various studies on legged locomotion, e.g., [13] and [19], suggest that legs store and dissipate energy during motion. Based on this evidence, we modify the passive SLIP by adding to the leg a piston-like actuator in series with the

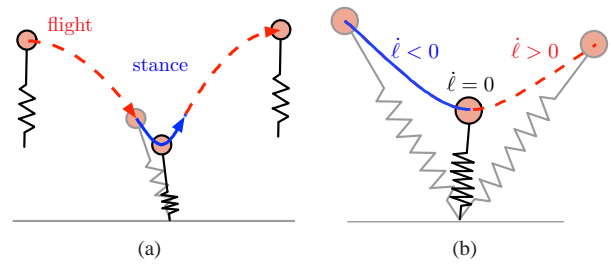


Fig. 2. (a) Scheme that illustrates the phases of the SLIP model trajectory, and (b) stance phase

spring, as shown in Fig. 1(b). We will refer to the actuated SLIP model as the *active SLIP*. Throughout the stance phase, the actuator can continuously extend or retract from its nominal position within a certain displacement range. Define  $\ell_{act}(t)$  to be the actuator length, and  $\ell_{act,0}$  be its nominal position at rest. Then, the equation that describes the evolution of the leg length (1) can be re-written as:

$$\ddot{\ell}(t) = -\frac{k}{M}(\ell(t) - \ell_0 - \ell_{act}(t)) - g \sin \theta + \ell \dot{\theta}^2. \quad (3)$$

The actuator manages absorption and production of energy during the stance phase by compressing and decompressing the spring, with the main advantage of allowing the energy at the beginning and at the end of the stance phase to be different.

Note that the take-off state happens at the time  $t_{TO}$  at which the spring reaches its equilibrium length,  $\ell_k(t_{TO}) = \ell_{k,0}$ . While for the passive SLIP this corresponds to the case  $\ell(t_{TO}) = \ell_0$ , the active SLIP will leave the ground when  $\ell(t_{TO}) = \ell_0 + \ell_{act}(t_{TO})$ .

## III. ACTUATION DISPLACEMENT: TWO-ELEMENT STRATEGY

### A. Approximating the stance phase dynamics through cancelling nonlinear terms

We propose a two-element strategy for the actuator displacement  $\ell_{act}(t)$  as follows. We divide the total actuator displacement in two parts:  $\ell_{act}(t) = \ell_{nl}(t) + \ell_c(t)$ , with total velocity  $v_{act}(t) = v_{nl}(t) + v_c(t)$ . The first term,  $\ell_{nl}(t)$ , has the purpose of cancelling the nonlinear terms in (3):

$$\ell_{nl}(t) = \frac{M}{k} [g \sin \theta(t) - \ell(t) \dot{\theta}(t)^2]. \quad (4)$$

We drive the second term,  $\ell_c$ , to a constant value. However, since the actuator does not move instantaneously, we assume that, after cancelling the nonlinearity, the actuator moves with a constant velocity  $v_c$  from its initial position until it reaches the desired value  $\ell_c$ :

$$\ell_{act}(t) = \begin{cases} \ell_{nl}(t) + v_c t, & \text{if } |v_c| t < |\ell_c| \\ \ell_{nl}(t) + \ell_c, & \text{otherwise.} \end{cases} \quad (5)$$

During the transition time required to reach the desired actuator value  $\ell_c$ , the equation that describes the leg-length dynamics can be analytically solved as

$$\ell(t) = \ell(t_i) + v_c t + a \cos(\omega t + \beta), \quad (6)$$

where<sup>1</sup>

$$\beta = \text{atan}_2\left(-\frac{\dot{\ell}(t_i) - v_c}{\omega}, \ell(t_i) - \ell_0 - v_c t_i\right), \quad a = -\frac{\dot{\ell}(t_i) - v_c}{\omega \sin \beta},$$

$t_i$  is the initial time, and  $\omega = \sqrt{k/M}$ .

Once the actuator reaches the desired final value  $\ell_c$ , the leg-length dynamics are described by

$$\ell(t) = \ell(t_c) + \ell_c + a \cos(\omega t + \beta), \quad (7)$$

where

$$\beta = \text{atan}_2\left(-\frac{\dot{\ell}(t_c)}{\omega}, \ell(t_c) - \ell_0 - \ell_c\right), \quad a = -\frac{\dot{\ell}(t_c)}{\omega \sin(\beta)},$$

and  $t_c$  is the time at which  $v_c t_c = \ell_c$ .

The equation of motion for the angular displacement over time,  $\theta(t)$ , is still not analytically solvable. However, we notice that eq. (7) has the same form as the approximation of the leg-length dynamics in [12], with two main differences: (i) [12] considers the passive SLIP model only, and (ii) the equation for the leg-length dynamics provided in [12] is an approximation, while eq. (7) and (6) are exact solutions. We then chose to follow the same initial steps of the procedure to approximate  $\theta(t)$  proposed in [12], modifying and extending the results to adapt them to our actuated case, as shown in the Appendix.

Note: in general,  $v_{nl}(t)$  is not a constant value. Since  $v_c$  is set to be a constant, the total actuator velocity required  $v_{act}(t)$  is a time-varying function.

#### B. Choice of the actuator constant value $\ell_c$

We now propose a strategy for the choice of the actuator constant value  $\ell_c$ .

Let us divide the stance phase in two parts, separated by the point of maximal leg compression: a first part, where  $\dot{\ell}(t) \leq 0$ , and a second part, where  $\dot{\ell}(t) \geq 0$ , as in Fig. 2(b). Our main control action consists in choosing two constant values for  $\ell_c$ : one for the first part,  $\ell_{c1}$ , and one for the second part  $\ell_{c2}$ , of the stance phase.

The decision of choosing only two constant values for the actuator displacement as opposed to a time dependent function (as, for example, in [16], [17], and [20]) has been dictated by the purpose of keeping the system as simple as possible, without much loss on performance. Fig. 3(a) and 3(b) provide an example of how, by setting only two actuator values, it is possible to reach in one step a wide range of apex states, influencing both apex height, apex velocity and apex forward position, in all directions.

#### IV. PERFORMANCE

To study the performance of our strategy for actuator displacement, we consider the non-dimensional *relative spring stiffness*,  $\gamma$ , defined as

$$\gamma = \frac{k\ell_0}{Mg}. \quad (8)$$

Simulations are conducted for  $\gamma \in [10, 200]$ , using constant values for  $\ell_0$ ,  $k$  and  $M$ . The initial apex height and velocity

<sup>1</sup>For any point  $(x, y)$  in the  $xy$ -plane except for the origin,  $\text{arctan}_2(y, x)$  is defined to be the angle between the horizontal positive axis and the point  $(x, y)$  measured counterclockwise.

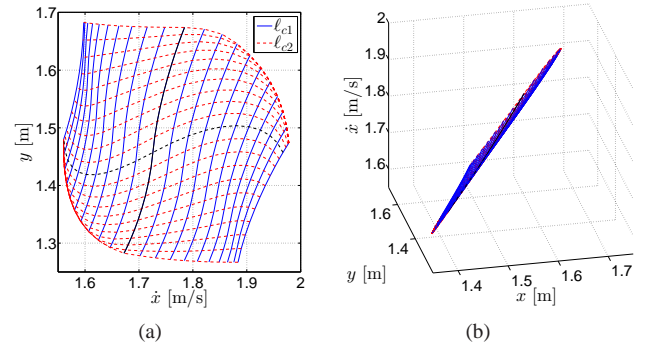


Fig. 3. Apex states  $\{y, \dot{x}\}$  (a), and  $\{x, y, \dot{x}\}$  (b), reachable in one step, with  $M = 10$  [kg],  $k = 1962$  [N/m],  $\ell_0 = 1$  [m]. Initial apex:  $x_{apex} = 0$  [m],  $y_{apex} = 1.4$  [m],  $\dot{x}_{apex} = 2$  [m/s],  $\theta_{TD} = 103$  [deg]. Maximum/minimum actuation length:  $\pm 0.05$  [m]. Actuator moves with velocity  $v_c = 0.5$  [m/s]. Each solid blue line corresponds to a different actuation value during the first half of the stance phase,  $\ell_{c1}$ , while each dotted red line corresponds to a different actuation value during the second half of the stance phase,  $\ell_{c2}$ . The black lines correspond to the case of either  $\ell_{c1} = 0$  or  $\ell_{c2} = 0$ .

have been chosen as a function of the leg length and the time scale  $\tau = 1$  [s] to be  $y_a \in [\ell_0, 2.5\ell_0]$  and  $\dot{x}_a \in [0.5\frac{\ell_0}{\tau}, 3\frac{\ell_0}{\tau}]$ , while touch-down angle has been chosen as  $\theta_{TD} \in [90, 150]$  [deg]. The spring length at equilibrium has been assumed to be  $\ell_{k,0} = 0.5\ell_0$ , with a maximum compression of  $\ell_{k,min} = 0.05\ell_0$ .

#### A. Feasibility

It is important to point out that, depending on the system's parameters and its initial conditions, the actuator displacement required to cancel the nonlinear terms,  $\ell_{nl}(t)$ , could exceed the maximum actuator displacement and velocity allowed, or could bottom-out the spring. For example, we can assume the total actuator displacement  $\ell_{act}(t) = \ell_c(t) + \ell_{nl}(t)$  not to exceed 10% of the leg length  $\ell_0$ , the maximum velocity  $v_{act} = v_c + v_{nl}$  not to exceed  $1\ell_0/\tau$ , and that  $\ell_{act}(t)$  at any given time would not bottom-out the spring, i.e.,  $\ell_k(t) \geq \ell_{k,min}$ . The amount of displacement and velocity for the nonlinear part and the constant part can be allocated in an infinite number of ways. For example, Fig. 4(a) shows the initial apex states that require  $\ell_{nl} \leq 0.5\ell_0$  and  $v_{nl} \leq 0.5\ell_0/\tau$  to perform a symmetric jump, while Fig. 4(b) considers apex states that require  $\ell_{nl} \leq 0.4\ell_0$  and  $v_{nl} \leq 0.3\ell_0/\tau$ . The simulations have been computed for several values of  $\gamma$ . As we can see, our strategy works well for values of  $\gamma \geq 100$  for both allocations considered. For  $\gamma \leq 10$  the size of the set of feasible initial conditions is small, posing a heavy limit to the application of our controlling strategy.

#### B. Error reduction

Now, we want to test the benefits of our proposed strategy for cancelling the nonlinear terms via our active SLIP control. We introduce the percentage errors of variables  $y$  and  $\dot{x}$ , respectively, as:

$$PE_y = 100 \frac{\|y - \tilde{y}\|_2}{\|y\|_2}, \quad PE_{\dot{x}} = 100 \frac{\|\dot{x} - \tilde{\dot{x}}\|_2}{\|\dot{x}\|_2}, \quad (9)$$

where  $\tilde{y}$  and  $\tilde{\dot{x}}$  are height and velocity at apex computed via approximation, while  $y$  and  $\dot{x}$  are the actual apex height and velocity.

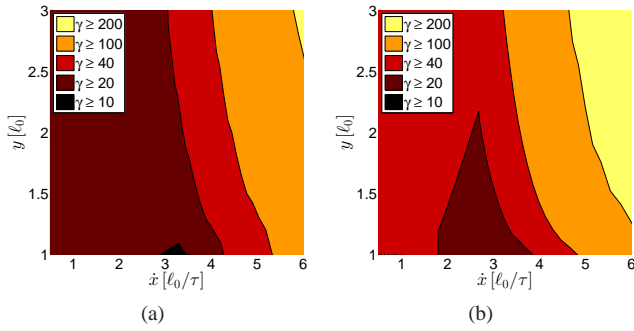


Fig. 4. Figure (a) shows the initial apex states that require  $\ell_{nl} \leq 0.5\ell_0$  and  $v_{nl} \leq 0.5\ell_0/\tau$  to perform a symmetric jump, while Figure (b) considers apex states that require  $\ell_{nl} \leq 0.4\ell_0$  and  $v_{nl} \leq 0.3\ell_0/\tau$ . Different shades refer to different intervals of  $\gamma$ :  $\gamma \geq 10$  (black),  $\gamma \geq 20$ ,  $\gamma \geq 40$ ,  $\gamma \geq 100$ , and  $\gamma \geq 200$  (light yellow).

and velocity computed using Matlab numerical solver *ode45*, with absolute and relative tolerances set at  $10^{-8}$ .

First of all, why is it useful to cancel the nonlinear terms, i.e., what is the benefit of having an exact solution for  $\ell(t)$ ? We answer this question by comparing the percentage errors (9) for the approximation proposed in [12] versus our approximation with nonlinearity cancel (4). Since the approximation in [12] does not consider actuation, the comparison is performed with respect to our approximation computed with  $\ell_{act}(t) = \ell_{nl}(t)$ , i.e.,  $\ell_c(t) = 0$ . Fig. 5(a) and 5(b) show the mean percentage errors  $PE_y$  and  $PE_{\dot{x}}$  for symmetric and non-symmetric trajectories, with values of  $\gamma \in [20, 200]$  (values of  $\gamma < 20$  have not been considered due to their limitation, as shown in Subsection IV-A). As

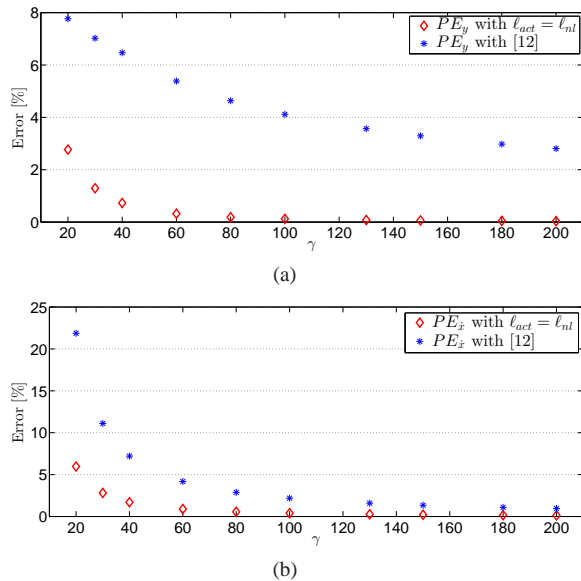


Fig. 5. The blue stars represent the PEs computed with respect to the stance phase approximation proposed in [12]. The red diamonds represent the PEs computed using our proposed approximation via nonlinearity cancellation. Initial apex conditions have been chosen to be  $y \in [\ell_0, 2.5\ell_0]$ ,  $\dot{x} \in [0.5\frac{\ell_0}{\tau}, 3\frac{\ell_0}{\tau}]$ , and  $\theta_{TD} \in [90, 150]$  [deg]. Maximum actuator displacement and velocity are  $\max(|\ell_{act}|) = 0.05\ell_0$  and  $\max(|v_{act}|) = 0.5\ell_0/\tau$ .

we can see, our proposed strategy significantly reduces the percentage errors, especially for lower values of  $\gamma$ . This can

serve as a starting point for the choice of  $\gamma$  while building a hardware prototype. Consistently, the hopper being under development in our Robotics Laboratory has  $\gamma \approx 20$ .

We now compute the percentage errors (9) for the actuated SLIP model, with our proposed actuation displacement strategy  $\ell_{act}(t)$  as in (5). Fig. 6(a) and 6(b) show the mean and standard deviations of, respectively, the percentage errors  $PE_y$  and  $PE_{\dot{x}}$ , computed for several values of  $\gamma \in [20, 200]$  and a set of 60,000 initial conditions.

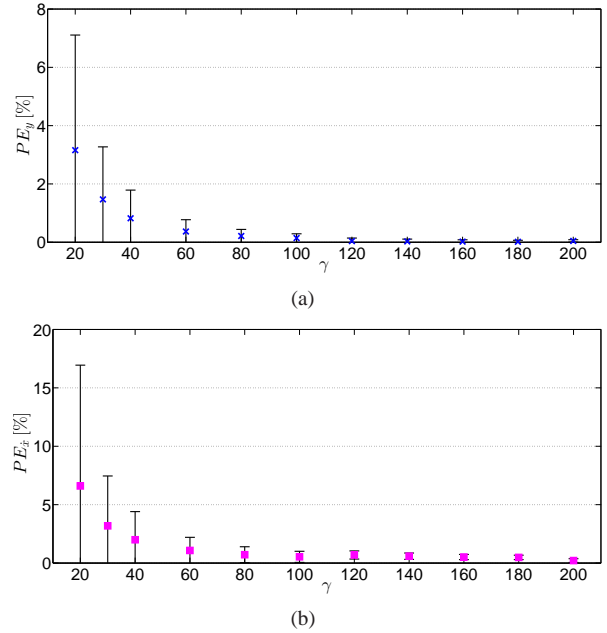


Fig. 6. Percentage errors  $PE_y$  (a) and  $PE_{\dot{x}}$  (b) for our proposed approximation with actuation displacement (5). The colored symbols and black lines represent respectively the mean values and the standard deviations of the percentage errors computed for a pool of 60,000 initial conditions. The initial apex conditions have been chosen to be  $y \in [\ell_0, 2.5\ell_0]$ ,  $\dot{x} \in [0.5\frac{\ell_0}{\tau}, 3\frac{\ell_0}{\tau}]$ , and  $\theta_{TD} \in [90, 150]$  [deg]. Maximum actuator displacement and velocity are  $\max(|\ell_{act}|) = 0.1\ell_0$  and  $\max(|v_{act}|) = 1\ell_0/\tau$ , with  $\ell_c \in [-0.05\ell_0, 0.05\ell_0]$ , and  $v_c \in [0.5\ell_0/\tau, 0.5\ell_0/\tau]$ . If either the total actuator displacement or velocity exceeds the maximum values allowed, the actuator is assumed to saturate during numerical computation of the stance phase trajectory.

## V. CONTROL ACTIONS

The main advantage of using an approximation of the stance phase versus its numerical solution is given by the reduction of computational time, which translates into the possibility of performing online control actions. To give an example, on a representative pool of 60,000 initial conditions (apex state and touch-down angle), we computed the average time to simulate the stance phase using Matlab's function *ode45* versus an analytical approximation. The calculations were performed on a Microsoft Windows based computer (Intel Core i7 eight core processor CPU, 2.80 GHz) using Matlab version R2012a. While the average time for *ode45* was 0.0259 [s], the average time for an approximate solution was  $9.4243 \times 10^{-5}$  [s]: a decrease in computation time of over 250 times.

We start from this preliminary remark to introduce our proposed control actions.

### A. Online Computation of Optimal Parameters

We use a modified version of the Matlab function *fminsearch* (which optimizes constrained problems using the Nelder-Mead algorithm). At any current apex state  $\mathbf{s}_n = \{y_n, \dot{x}_n\}$ , we compute the values for the touch-down angle  $\theta_{TD}$  and the two actuator values  $\ell_{c1}$  and  $\ell_{c2}$  that minimize in one step the distance to a desired apex  $\mathbf{s}_{des} = \{y_{des}, \dot{x}_{des}\}$ . The optimization problem is defined to be constrained due to the bounds on the values taken by the touch-down angle and the actuator displacement.

The cost function to minimize,  $J$ , is defined as:

$$J = 100 \sqrt{\frac{(y_{n+1} - y_{des})^2}{y_{des}^2} + \frac{(\dot{x}_{n+1} - \dot{x}_{des})^2}{\dot{x}_{des}^2}}, \quad (10)$$

which expresses the percentage distance from the next apex state  $\mathbf{s}_{n+1} = \{y_{n+1}, \dot{x}_{n+1}\}$  to the desired one,  $\mathbf{s}_{des}$ .

Note that the average time to solve the optimization problem has been computed to be 0.05 [s], which is much faster than the average ballistic apex-to-apex time. In particular, the shortest apex-to-apex time corresponds to a take-off and touch-down angle  $\theta_{TO} = 90$  [deg] and  $\theta_{TD} = 90$  [deg]. In order to guarantee the apex-to-apex time to be smaller than 0.05 [s], it is required for the apex height to be  $y_{ap} \geq 1.0123\ell_0$ .

### B. Adaptive control for steady-state achievement

Due to errors in the approximation (see Fig. 6(a) and 6(b)), the touch-down angle and actuator values that minimize the cost function (10) may drive the system to an apex state distant from the desired one by a certain percentage error. Therefore, the system will converge to a steady-state apex state  $\mathbf{s}_{ss}$  that is not the desired one. In order to avoid this problem, we propose here a strategy to reduce such steady state errors, driving the system closer to the desired steady state over time. The strategy is summarized as follows.

Let us start from an initial apex state,  $\{y(0), \dot{x}(0)\}$ , and let us assume we want to ultimately reach the value  $\{y_{des}, \dot{x}_{des}\}$ . If for 2 consecutive steps the system reaches the same value (within a certain percentage error), we call that value the steady state value. When at the  $n$ -th step the system reaches the steady-state apex, we update the desired value for the next step  $y_{des}(n+1)$  and  $\dot{x}_{des}(n+1)$  to be

$$\begin{aligned} y_{des}(n+1) &= y_{des}(n) - y(n) + y_{des}, \\ \dot{x}_{des}(n+1) &= \dot{x}_{des}(n) - \dot{x}(n) + \dot{x}_{des}. \end{aligned}$$

Fig. 7(a) and 7(b) show the percentage distance  $J$  from the desired apex state after 6 steps on flat terrain for  $\gamma = 20$ , respectively (a) without and (b) with our steady-state adaptive control. Our proposed controller reduces the percentage distance  $J$  after 6 steps from a maximum of about 18% to a maximum of about 2.5%.

## VI. APPLICATIONS

In this section we show the performance of our controller to perturbations in terrain. In particular, we test recovery from perturbations on the terrain height, and we show an example of running on rough terrain.

The parameters and initial conditions used in our simulations are defined in Table I, and are based on biological

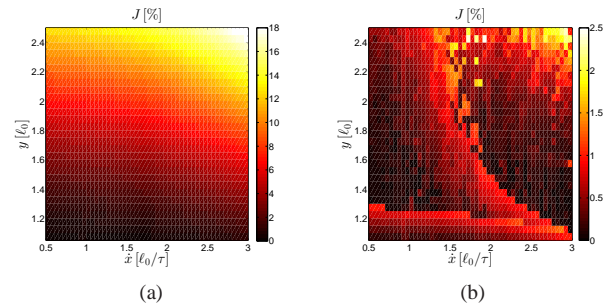


Fig. 7. These plots show the percentage error  $J$  after 6 steps, for  $\gamma = 20$ . The x-axis and y-axis represent the apex velocity and the apex height, respectively. Plot (a) refers to the case without our proposed steady-state achievement control, while plot (b) refers to the controlled case.

data for a typical human. Results in [2] show that  $\gamma \simeq 10$  for a hopping monopod, and  $\gamma \simeq 20$  for a hopping biped. Because of the limitations shown in Fig. 4(a) and 4(b), for our simulations we use  $\gamma = 20$ , which also matches a hardware prototype currently under development in our lab. During simulations, if either the total actuator displacement  $\ell_{act} = \ell_{nl} + \ell_c$  or the total actuator velocity  $v_{act} = v_{nl} + v_c$  required were exceeding the maximum values allowed, the actuator was assumed to saturate its maximum allowed value for  $\ell_{act}$  or  $v_{act}$ , respectively. Furthermore, to acknowledge the time to solve the optimization problem, we limit our simulations to initial apex heights and touch down angles that corresponds to a time during flight  $t_f \geq 0.15$  seconds.

Simulation Parameters	
$g =$	9.81 m/s <sup>2</sup>
$M =$	80 kg
$\ell_0 =$	1 m
$\ell_{k,0} =$	0.5 m
$\ell_{k,min} =$	0.05 m
$\ell_{act} \in$	[-0.1, 0.1] m
$\ell_c \in$	[-0.05, 0.05] m
$v_{act} \leq$	1 m/s
$v_c =$	0.5 m/s
$y \in$	[1, 2.5] m
$\dot{x} \in$	[0.5, 3] m/s
$\gamma =$	20

TABLE I

### A. Recovery from perturbations

We considered the set of initial conditions in Table I. We test the recovery capabilities of our controller when the active SLIP encounters an unexpected (positive or negative) perturbation on the terrain height of up to 60% of the leg length  $\ell_0$ .

At each apex state, we use the strategy in V-A to compute the optimal values for  $\ell_{c1}$ ,  $\ell_{c2}$  and  $\theta_{TD}$  for flat terrain. Once the leg touches the ground with the computed touch-down angle and the desired  $\ell_{c1}$ , the strategy in V-A is simulated again during the first half of the stance phase, this time to compute only  $\ell_{c2}$  to take into account the encountered perturbation on the terrain height. Note that the search of this second value takes on average 0.01 [s], which is much smaller than the average time required for the first half of the stance phase, and therefore it can be realistically implemented. Fig. 8(a), 8(b), 8(c), and 8(d) show the number

of steps necessary for the system in order to return within 1% of the initial apex state, in the case of positive or negative perturbations. As we can see, our controller is robust to perturbations of varying magnitudes, with ability to recover in up to 12 steps. Fig. 9(a) and 9(b) show an example of

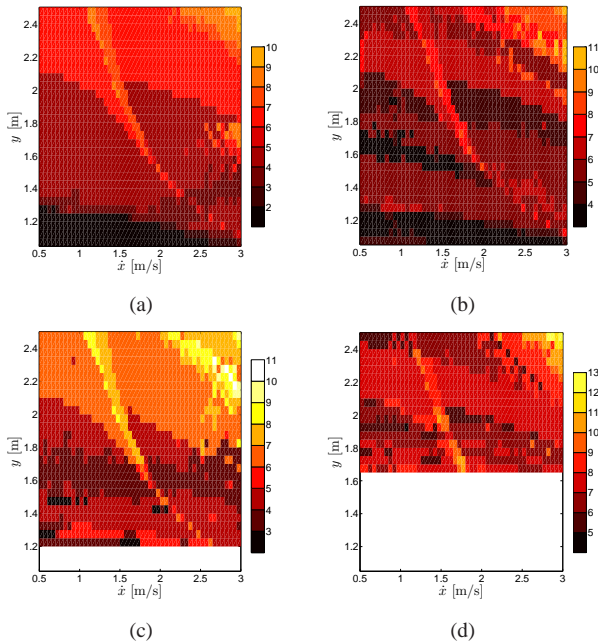


Fig. 8. Number of steps to reach 1% of desired value, for terrain perturbation of magnitude (a)  $-0.2$  [m], (b)  $-0.6$  [m], (c)  $0.2$  [m], (d)  $0.6$  [m]. Simulation parameters are chosen as per Table I. Note that in case of a positive terrain perturbation (figure (c) and (d)), the apex height  $y$  has been chosen to leave enough room to the leg to swing during flight without colliding with the terrain, thus explaining the empty areas at the bottom of the figures.

apex recovery for an unforeseen drop of magnitude 60% of the leg length.

### B. Hopping on rough terrain

In this subsection we show an example of the active SLIP model hopping on rough terrain. Additionally, we assume that the terrain measurements are faulty, and we want to maintain the same forward velocity and the same distance from the terrain with respect to the last step. As we can see from Figure 10(a) and 10(b), the system is able to successfully hop on a rough terrain with a maximum magnitude landing height of  $0.43$  [m] and a maximum perturbation of  $0.37$  [m], i.e., 43% and 37% of the leg length, respectively. Consistent with what shown on Fig. 6(a) and 6(b), the error on  $\dot{x}$  is on average higher than the error on  $y$ , and in this example they both do not exceed 5%. Note that, since the terrain height varies continuously, the energy of the system varies at each step.

## VII. CONCLUSIONS AND FUTURE WORK

In this paper, we considered the active SLIP model, a non-conservative extension of the classic SLIP that add an actuator in series with the spring. We proposed a two-part strategy for actuator displacement to obtain an analytical solution to the equation that describes the leg-length dynamics during stance, and we provide an approximation of the leg-angle

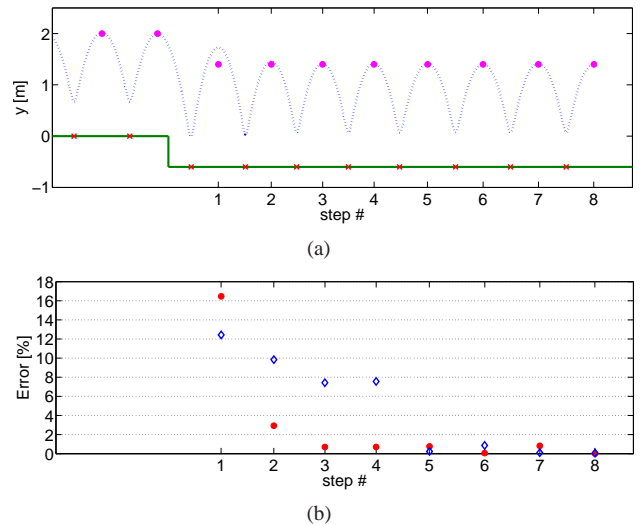


Fig. 9. Figure (a) shows the trajectory of the active SLIP for initial apex state  $y = 2$  [m],  $\dot{x} = 1.3$  [m/s], and negative terrain perturbation of magnitude  $0.6$  [m]. The blue dotted line represent the trajectory of the mass, the purple circles the desired apex height. The green solid line is the terrain height and the red crosses are the landing points of the foot. Figure (b) shows the percentage error of apex height (red circle) and velocity (blue diamond) at each step after the drop. As we can see, the controller reaches and remains within 1% of the original apex state in 5 steps.

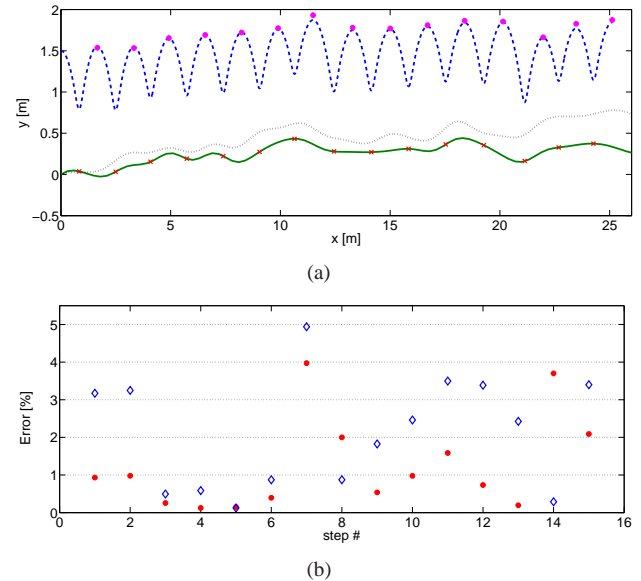


Fig. 10. Figure (a) shows the trajectory of the active SLIP hopping on a random-generated rough terrain. The controller acts to maintain a constant apex height with respect to the terrain of  $y = 1.5$  [m], and a constant forward velocity of  $\dot{x} = 2$  [m/s]. The blue dotted line represent the trajectory of the mass, the purple circles the desired apex height. The black dotted line is the expected terrain height, while the green solid line is the actual terrain height and the red crosses are the landing points of the foot. Fig.(b) shows the percentage error of apex height (red circle) and velocity (blue star) at each step.

dynamics. Additionally, we develop a control strategy for actuator displacement that allows us to add/remove energy from the system to drive it to a desired apex state, and we validated it through simulations, even in the presence of perturbations on the terrain height. Furthermore, we provided a feasibility study of our strategy for different system parameters.

We are currently extending our work in several directions. First, we want to extend our actuator displacement strategy and stance-phase approximation to take into account actuator displacement with finite acceleration. Second, we want to improve our adaptive control for steady-state achievement to obtain convergence to the desired apex state. Third, we aim to adapt our modelling approach to include energy loss at touch-down, leg inertia, and generally to more accurately match a real hardware prototype; and to verify our theoretical ideas on a hopping robot currently under development in our lab.

#### ACKNOWLEDGMENTS

This work is supported by DARPA (Grant No. W911NF-11-1-0077).

#### REFERENCES

- [1] R. Blickhan, "The spring-mass model for running and hopping," *Journal of Biomechanics*, vol. 22, no. 11/12, pp. 1217–1227, 1989.
- [2] 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, no. 5, pp. 509–517, Nov. 1993.
- [3] C. T. Farley and D. P. Ferris, "Biomechanics of walking and running: Center of mass movements to muscle action," *Exercise and Sport Science Reviews*, vol. 26, pp. 253–283, 1998.
- [4] M. H. Raibert, *Legged Robots that Balance*. MIT Press, 1986.
- [5] G. Zeglin, "The Bow Leg Hopping Robot," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, USA, Oct. 1999.
- [6] J. W. Hurst, J. Chestnutt, and A. Rizzi, "Design and Philosophy of the BiMASC, a Highly Dynamic Biped," in *IEEE Int. Conf. on Robotics and Automation*, Roma, Italy, Apr. 2007.
- [7] M. Ernst, H. Geyer, and R. Blickhan, "Spring-Legged Locomotion on Uneven Ground: a Control Approach to Keep the Running Speed Constant," in *12th International Conference on Climbing and Walking Robots (CLAWAR)*, 2009, pp. 639–644.
- [8] Ö. Arslan and U. Saranlı, "Reactive Planning and Control of Planar Spring-Mass Running on Rough Terrain," *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 567–579, 2012.
- [9] 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.
- [10] H. Geyer, A. Seyfarth, and R. Blickhan, "Spring-mass running: simple approximate solution and application to gait stability," *Journal of Theoretical Biology*, vol. 232, pp. 315–328, Feb. 2005.
- [11] Ö. Arslan, U. Saranlı, and Ö. Morgül, "Approximate Stance Map of the Spring Mass Hopper with Gravity Correction for Nonsymmetric Locomotions," in *IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan, May 2009, pp. 2388–2393.
- [12] H. Yu, M. Li, and H. Cai, "Approximating the Stance Map of the SLIP Runner Based on Perturbation Approach," in *IEEE Int. Conf. on Robotics and Automation*, Saint Paul, MN, USA, May 2012, pp. 4197–4203.
- [13] D. Dudek and R. J. Full, "Passive mechanical properties of legs from running insects," *Journal of Experimental Biology*, vol. 209, pp. 1502–1515, 2006.
- [14] M. A. Daley, G. Felix, and A. A. Biewener, "Running stability is enhanced by a proximo-distal gradient in jointy neuromechanical control," *Journal of Experimental Biology*, vol. 210, pp. 383–394, 2007.
- [15] 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.
- [16] J. Schmitt and J. Clark, "Modeling Posture-Dependent Leg Actuation in Sagittal Plane Locomotion," *Bioinspiration and Biomimetics*, vol. 4, pp. 1–17, 2009.
- [17] G. Piovvan and K. Byl, "Enforced Symmetry of the Stance Phase for the Spring-Loaded Inverted Pendulum," in *IEEE Int. Conf. on Robotics and Automation*, Saint Paul, MN, USA, May 2012, pp. 1908–1914.
- [18] K. Byl, M. Byl, M. Rutschmann, B. Satzinger, L. van Blarigan, G. Piovvan, and J. Cortell, "Series-Elastic Actuation Prototype for Rough Terrain Hopping," in *IEEE International Conference on Technologies for Practical Robot Applications*, 2012, pp. 103–110.

- [19] T. Lejeune, P. Willems, and N. Heglund, "Mechanics and energetics of human locomotion on sand," *Journal of Experimental Biology*, vol. 201, pp. 2071–2078, 1998.
- [20] M. Rutschmann, B. Satzinger, M. Byl, and K. Byl, "Nonlinear Model Predictive Control for Rough Terrain Hopping," to be presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems, October 2012.

#### APPENDIX

##### A. Approximation of the leg angle dynamics $\theta(t)$ during stance

When the actuator reaches the final desired value  $\ell_c$ , the equation that describe the dynamics of the leg length during stance is Eq. (7). We notice that this equation has the same form as the approximation of the leg length dynamics proposed in [12], with the difference that, in our case, the solution is exact and it includes the actuator displacement. To approximate  $\theta(t)$  we then chose to follow the same procedure proposed in [12]. We obtain

$$\theta(t) = \frac{1}{\ell(t)}(u_0(t) + \epsilon u_1(t)), \quad (11)$$

where  $u_0(t)$  and  $u_1(t)$  are solutions of

$$\begin{aligned} \frac{d^2}{dt^2}u_0 - \lambda^2 u_0 &= 0, \\ \frac{d^2}{dt^2}u_1 - \lambda^2 u_1 &= -\delta u_0 \cos \omega t + \beta, \end{aligned}$$

and

$$\epsilon = a/r, \quad \delta = \omega^2 + g/r, \quad \lambda^2 = g/r.$$

During the transition time required to reach the desired value  $\ell_c$ , the equation that describes the leg length dynamics can be analytically solved as (6). We want then to modify the procedure to approximate the leg length dynamics proposed in [12] to adapt it to (6). Equation (35) in [12] becomes:

$$\frac{d^2 u}{dt^2} - u(t) \frac{c_2 \omega^2 \cos \omega t + \beta + g}{c_1 + c_2 \cos \omega t + \beta + v_c t}.$$

Defining  $\psi = \omega t + \beta$ , we have  $\frac{d}{dt}u(t) = \omega u'(\psi)$  and  $\frac{d^2}{dt^2}u(t) = \omega^2 u''(\psi)$ , where  $(\cdot)'$  represents the derivative with respect to  $\psi$ . Then, the previous equation becomes:

$$u''(\psi) - u(\psi) \left( -1 + \frac{1 + \zeta + \xi \psi}{1 + \kappa \cos \psi + \xi \psi} \right) = 0, \quad (12)$$

with

$$\xi = \frac{v_c}{c_1 \omega - \beta v_c}, \quad \kappa = \frac{\xi c_2 \omega}{v_c}, \quad \zeta = \frac{\xi g}{\omega v_c}.$$

We can expand the fractional term in (12) around  $\cos \psi = 0$ , obtaining

$$\begin{aligned} -1 + \frac{1 + \zeta + \xi \psi}{1 + \kappa \cos \psi + \xi \psi} &\simeq \\ \tilde{\lambda}^2 - \tilde{\delta}(\tilde{\epsilon} \cos \psi - \tilde{\epsilon}^2 \cos^2 \psi + \tilde{\epsilon}^3 \cos^3 \psi - \dots). \end{aligned}$$

Then

$$\theta(t) = \frac{1}{\ell(t)}(\tilde{u}_0(t) + \tilde{\epsilon} \tilde{u}_1(t)),$$

where  $\tilde{u}_0(t)$  and  $\tilde{u}_1(t)$  are solutions of

$$\begin{aligned} \tilde{u}_0' - \tilde{\lambda}^2 \tilde{u}_0 &= 0, \\ \tilde{u}_1'' - \tilde{\lambda}^2 \tilde{u}_1 &= -\tilde{\delta} \tilde{u}_0 \cos \psi. \end{aligned}$$