Partial Feedback Linearization and Control of the SLIP Model Via Two-Element Leg Actuation Strategy

Giulia Piovan and Katie Byl

***Abstract*—The Spring Loaded Inverted Pendulum (SLIP) has been extensively studied and used as an inspiration to research on 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 added series actuation. In particular, we propose a two-element control strategy for actuator displacement to add/remove energy from the system, and analytically solve part of its dynamics. In addition, we develop two control strategies for online computation of actuator displacement and leg positioning: one to drive the system to a desired state, even in the presence of terrain perturbation; the other to control the system to hop on a desired set of terrain footholds. Furthermore, we propose an adaptive control technique for steady-state locomotion on flat terrain to reduce approximation errors by the use of an approximation of the leg-angle dynamics during the stance phase.**

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 to produce or remove 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 [2], the classic SLIP has been a fundamental tool to model running behavior for a variety of animals in biomechanical studies (e.g, [3] and [4]), as well as robotics. Perhaps one of the first and most famous SLIP-based one-legged robots is the Raibert hopper [5], and several other robots have also been developed based on the one-legged bouncing model (e.g, [6], [7], [8], and [9]). Consequently, various control strategies have been proposed. Some studies consider stable periodic gaits: e.g, [10] prove that setting a specific fixed touch-down angle produces asymptotically stable periodic gaits, and [11] investigates stability of vertical hopping while varying spring stiffness and leg length. Other studies focus on negotiating uneven terrains: e.g., [12] propose a control action to keep the running speed constant, [13], show simulated and experimental results of three main control strategies applied to a single-legged hopping robot on unknown terrain, while [14] propose an algorithm for trajectory planning, robust to model uncertainty and measurement noise. In [15] and [16], a swing-leg retraction strategy and its optimal rate have been proposed, respectively. One of the main limitations 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., [17], [18], [19], [20]). 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 non-negligible, with the consequent need of an improved approximation.

Another interesting issue arises from biological data: studies on running insects [21] and guinea fowls [22] 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. There is clearly a need to examine the effect of adding actuations to the system to manage energy variations. In [23], 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 have been investigated in [24], [25], and [26]. In [24], the authors consider energy variation during stance to produce asymptotically stable gaits, while [25] focus on a control action for actuator displacement that allows for an analytic solution to the stance phase equations. Work in [27] combines an actuator displacement strategy with a model predictive control approach to plan a trajectory based on a fixed set of desired footholds.

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

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

An initial version of this work appeared as [1]; differences with this article include a more detailed control action section, with the addition of a foothold placement control. Additionally, we ameliorate the adaptive control strategy for steady-state locomotion, we provide a more detailed computation of the proposed partial approximation of the stance phase, we extend the application section, and fix a few typos that were present in the preliminary version.

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, or to a desired set of terrain footholds. 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 height and forward velocity, paired with an adaptive control technique to reduce bias on error on flat terrain. A strategy for foothold placement via model predictive control is also introduced, and compared to an existing one. Section VI shows the performance of the proposed control actions with perturbation on terrain. Finally, Section VII contains conclusions and future work.

II. SLIP Model: System Dynamics

In this section we review the structure of the passive SLIP model and its dynamics. We also introduce its actuated version that will be used throughout the paper, the so-called active SLIP model, explaining in detail its dynamics and the modifications made to the original model to incorporate energy variations.

A. Passive SLIP Model

The passive SLIP is modelled as a point mass, M, attached to a massless spring leg, with length ℓ and spring stiffness constant k. Running dynamics for the SLIP model occur on the sagittal plane, and constitute an hybrid system. In fact, they 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/extension of the spring completely defines the mass dynamics. 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. During the flight phase gravity is the only force acting on the system. Defining as x and y respectively the forward and vertical coordinates of the mass, the equations of motion during flight can be written as:

$$\ddot{x}(t) = 0,$$
$$\ddot{y}(t) = -g,$$

where, as customary, g is the gravitational acceleration. 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 $\boldsymbol{s} = \{x_a,\, y_a,\, \dot{x}_a\}$.

The stance phase starts with the leg hitting the ground with a touch-down angle θ_{TD}. While the body moves forward, the spring compresses until it reaches its minimum compression point, then it starts expanding. When the spring reaches its equilibrium position (i.e., when the forces in the spring are back to zero), the system leaves the ground with a certain take-off angle θ_{TO}. As shown in Fig. 1, 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 ℓ_k is the spring length, and $\ell_{k,0}$ is the spring length at equilibrium. The state of the mass can be easily converted from Cartesian coordinates into polar coordinates:

$$x(t) = \ell(t)\cos\theta(t), \quad y(t) = \ell(t)\sin\theta(t).$$

Then, the Lagrangian for the system during stance phase is derived as follows:

$$L = \frac{M}{2}(\ell^2\dot{\theta}^2 + \dot{\ell}^2) - Mg\ell\sin\theta - \frac{k}{2}(\ell_k - \ell_{k,0})^2,$$

and the equations of motion in polar coordinates for the stance phase can be written as:

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

$$\ddot{\theta} = -2\frac{\dot{\ell}}{\ell}\dot{\theta} - \frac{g}{\ell}\cos\theta. \tag{2}$$

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

In this work, we will call *jump* the transition from one apex state to the next. A jump is *successful* if the forward velocity during the entire trajectory only takes positive values. Furthermore, to allow the leg to swing forward in flight, we require the distance between y_a and the terrain height to be bigger than the leg length at equilibrium ℓ_0.

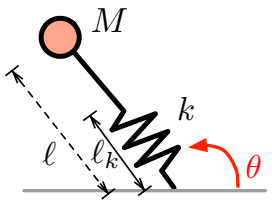


Fig. 1. Classic SLIP model (passive SLIP)

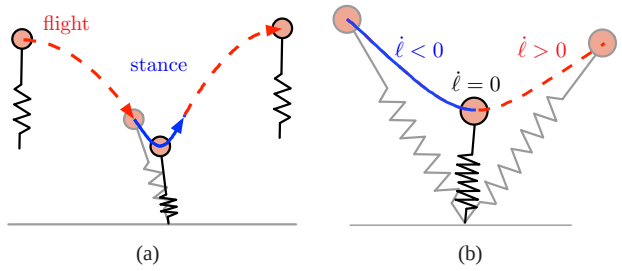


Fig. 2. (a) Scheme that illustrates the phases of the SLIP model trajectory, and (b) stance phase

B. Active SLIP model

The classic SLIP model is energetically conservative: the repositioning of the leg to the desired touch-down angle does not require any energy, and no energy is lost during impact with the ground as well. However, various studies on legged locomotion, e.g., [21] and [28], 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 spring, as shown in Fig. 3. We will refer to the actuated SLIP model as the *active SLIP*. Throughout the stance phase, the actuator can continuously

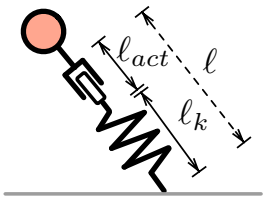


Fig. 3. Active SLIP model (active SLIP)

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:

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

where $\ell_0 = \ell_{act,0} + \ell_{k,0}$. 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. Positive and negative values of ℓ_{act} correspond to a compression and extension of the spring, respectively.

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 partial feedback linearization

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, $\ell_{nl}(t)$, performs a partial feedback linearization: it has the purpose of cancelling the nonlinear terms in (3):

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

Note that, at touch-down, this becomes:

$$\ell_{nl,TD} = \frac{M}{k}\big(g\sin\theta_{TD} - \ell_0\dot{\theta}_{TD}^2\big),$$

which is in general not zero, and therefore the spring might need to be pre-compressed or pre-extended during flight. Substituting (4) in (3), we obtain:

$$\ddot{\ell}(t) = -\frac{k}{M}(\ell(t) - \ell_0 - \ell_c(t)),$$

where $\ell_c(t)$ is chosen as follows.

We drive the second term, $\ell_c(t)$, to a constant value $\bar{\ell}_c$. 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 $\bar{\ell}_c$:

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

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.

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

$$\ell(t) = r + v_c t + a\cos(\omega t + \beta), \quad (5)$$

where[1]

$$\beta = \text{atan}_2\Big(-\frac{\dot{\ell}(t_i) - v_c}{\omega},\ \ell(t_i) - \ell_0 - \ell_c(t_i)\Big),$$
$$a = \frac{\ell(t_i) - \ell_0 - \ell_c(t_i)}{\cos\beta}, \quad r = \ell_0 + \ell_c(t_i),$$

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

Once the actuator reaches the desired final value $\bar{\ell}_c$, the leg-length dynamics are described by

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

where

$$\beta = \text{atan}_2(-\frac{\dot{\ell}(t_c)}{\omega},\ \ell(t_c) - \ell_0 - \bar{\ell}_c),$$
$$a = \frac{\ell(t_c) - \ell_0 - \bar{\ell}_c}{\cos(\beta)}, \quad r = \ell_0 + \bar{\ell}_c,$$

and t_c is the time at which $v_c t_c = \bar{\ell}_c$.

The equation of motion for the angular displacement over time, $\theta(t)$, is still not analytically solvable. However, we notice that equation (6) has the same form as the approximation of the leg-length dynamics in [19], with two main differences: (i) [19] considers the passive SLIP model only, and (ii) the equation for the leg-length dynamics provided in [19] is an approximation, while (6) and (5) are exact solutions. We then chose to follow the same initial steps of the procedure to approximate $\theta(t)$ proposed in [19], modifying and extending the results to adapt them to our actuated case.

When the actuator reaches the final desired value ℓ_c, the equation that describes the dynamics of the leg length during stance is (6). Let us define $\alpha(t) = \theta(t) - \pi/2$, and assume that

[1]For any point (x, y) in the xy-plane minus the origin, $\arctan_2(y, x)$ is defined to be the angle between the horizontal positive axis and the point (x, y) measured counterclockwise.

the angular span $\Delta\alpha = \alpha_{TD} - \alpha_{TO}$ is sufficiently small. This yields $\sin(\alpha) \approx \alpha$. Therefore, equation (2) can be simplified and re-written in terms of α, becoming:

$$\ddot{\alpha}(t) = -2\frac{\dot{\ell}(t)\dot{\alpha}(t)}{\ell(t)} + \frac{g}{\ell(t)}\alpha(t). \tag{7}$$

We make the substitution $\alpha(t) = p(t)u(t)$ in (7), and obtain

$$\ddot{u}p + \dot{u}(2\dot{p} + 2\frac{\dot{\ell}p}{\ell}) + u(\ddot{p} + 2\frac{\dot{\ell}\dot{p}}{\ell} - g\frac{p}{\ell}) = 0.$$

Setting the coefficients associated to $\dot{u}$ to be equal to zero yields

$$\ell\dot{p} + \dot{\ell}p = 0,$$

whose solution is $p = 1/\ell$. Then, equation (7) becomes:

$$\ddot{u} - u\big(\frac{\ddot{\ell} + g}{\ell}\big) = 0. \tag{8}$$

When the actuator reaches the final desired value ℓ_c, the equation that describe the dynamics of the leg length during stance is equation (6). Then, (8) becomes

$$\ddot{u} - u\left(-\omega^2 + \frac{\omega^2 + g/r}{1 + \epsilon\cos(\omega t + \beta)}\right) = 0,$$

with $r = \ell(t_c) + \ell_c$, and $\epsilon = \frac{a}{r}$. We can write

$$\epsilon = \frac{z}{\cos\beta},$$

where

$$z = \frac{\ell(t_c)}{\ell_0 + \bar{\ell}_c}, \quad \text{or} \quad z = \frac{\ell(t_i)}{\ell_0 + \ell_c(t_i)}.$$

Let us assume small leg compression and small actuation, i.e., $(\ell(t) - \ell_0 - \ell_c)/(\ell_0 + \ell_c) \ll 1$, then we have that

$$\lim_{z\to 0}|\epsilon| = |\frac{\dot{\ell}(t_c)}{\omega(\ell_0 + \bar{\ell}_c)}|,$$

which, for typical values of w and leg length velocity, has magnitude less than 1. Then, for small values of ϵ, we have that

$$\frac{1}{1 + \epsilon\cos(\omega t + \beta)} \approx 1 - \epsilon\cos(\omega t + \beta) + \epsilon^2\cos^2(\omega t + \beta) - \cdots,$$

and thereby obtain the Mathieu equation

$$\ddot{u} - u\left(\lambda^2 - \epsilon\delta\cos(\omega t + \beta)\right), \tag{9}$$

with

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

Equation (9) can be developed as

$$u(t) = u_0(t) + \epsilon u_1(t) + \epsilon^2 u_2(t) + \cdots.$$

We then obtain

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

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

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

We want now to modify the procedure to approximate the leg length dynamics proposed in [19] and performed above, to adapt it to (5). During the transition time required to reach the desired value ℓ_c, the equation that describes the leg length dynamics can be analytically solved as (5). Then, equation (8) becomes

$$\ddot{u}(t) - u(t)\frac{-c_2\omega^2\cos(\omega t + \beta) + g}{c_1 + c_2\cos(\omega t + \beta) + v_c t} = 0.$$

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

$$u''(\psi) - u(\psi)\Big(-1 + \frac{1 + \zeta + \xi\psi}{1 + \kappa\cos\psi + \xi\psi}\Big) = 0, \tag{10}$$

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 (10) around $\cos\psi = 0$, obtaining

$$-1 + \frac{1 + \zeta + \xi\psi}{1 + \kappa\cos\psi + \xi\psi} \backsimeq$$
$$\tilde{\lambda}^2 - \tilde{\delta}(\tilde{\varepsilon}\cos\psi - \tilde{\varepsilon}^2\cos^2\psi + \tilde{\varepsilon}^3\cos^3\psi - \cdots),$$

where

$$\tilde{\varepsilon} = \frac{2(\kappa - \xi)}{2 + \pi\xi}, \quad \tilde{\lambda}^2 = \frac{2\zeta}{2 + \pi\xi}, \quad \tilde{\delta} = \frac{\kappa}{(\kappa - \xi)} + \frac{2(1 + \zeta)}{2 + \pi\xi}.$$

Then

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

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

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

B. Choice of the actuator constant value ℓ_c

We now propose a strategy for the choice of the actuator constant value ℓ_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 $\dot{\ell}(t) \geq 0$.. Our main control action consists in choosing two constant values for ℓ_c: one for the first part, ℓ_{c1}, and one for the second part ℓ_{c2}, of the stance phase, as shown in Fig. 4.

The decision of choosing only two constant values for the actuator displacement as opposed to a time dependent function (as, for example, in [24], [25], and [27]) has been dictated by the purpose of keeping the system as simple as possible, without much loss on performance. Fig. 5(a) and 5(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.

As stated earlier, moving the actuator results in a change of the system's energy. The energy at apex is defined as

$$E_a = Mgy_a + \frac{1}{2}M\dot{x}_a^2. \tag{11}$$

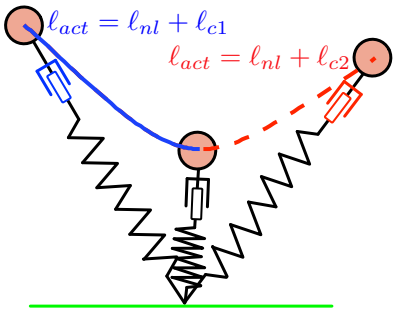
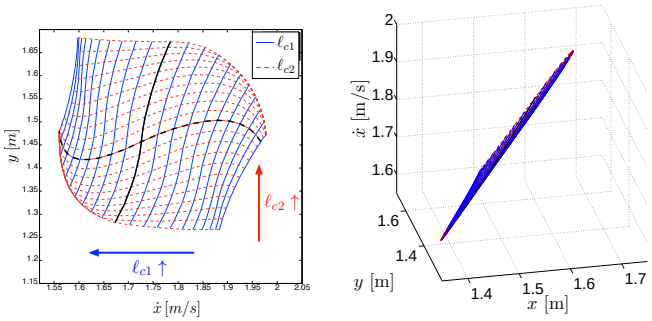


Fig. 4. Actuator constant value $\bar{\ell}_c$ during stance phase



(a) (b)

Fig. 5. 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: ± 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, ℓ_{c1}, while each dotted red line corresponds to a different actuation value during the second half of the stance phase, ℓ_{c2}. The black lines correspond to the case of either $\ell_{c1} = 0$ or $\ell_{c2} = 0$.

Fig. 6 shows the change in energy from one initial apex state to the next with different values for the couple $(\ell_{c1},\ \ell_{c2})$. As a general rule of thumb, the maximum energy increase is obtained by extending the spring in the first half of the stance phase (i.e., $\ell_{c1} < 0$) and compressing it during the second half (i.e., $\ell_{c2} > 0$), while compressing first, and then extending (i.e., $\ell_{c1} > 0$ and $\ell_{c2} < 0$) results in a maximum energy decrease.

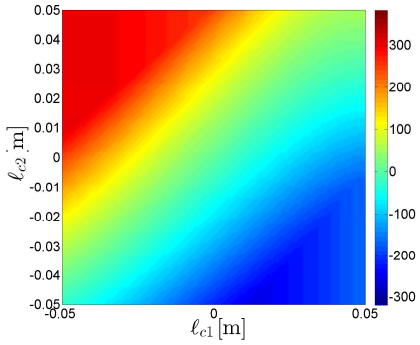


Fig. 6. The colorbar represents the energy change values $\Delta E = E_{a,2} - E_{a,1}$. $E_{a,1}$ is the energy (11) at the initial apex state $y_a = 1.4$ [m] and $\dot{x}_a = 2$ [m/s]. $E_{a,2}$ is the energy (11) at each next apex state computed for $\theta_{TD} = 103$ [deg] and varying values of the couple $(\ell_{c1},\ \ell_{c2})$.

IV. Performance

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

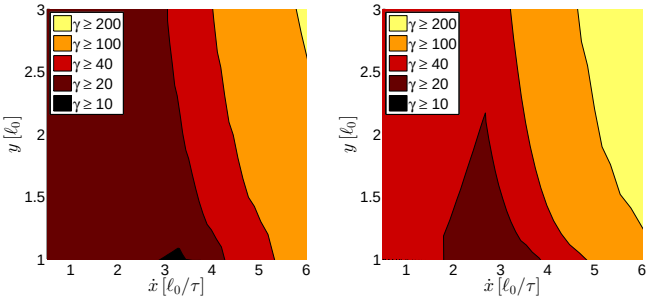
$$\gamma = \frac{k\ell_0}{Mg}. \tag{12}$$

The introduction of the relative spring stiffness is motivated by the fact that, if we focus on biped and quadruped robots where each leg functions and behaves the same way as the other(s), we can study their performance by considering the relative spring stiffness γ per leg (or net per multiple legs in contact). We can then study the performance of our monoped hopper in terms of γ.

Simulations are conducted for $\gamma \in [10,\ 200]$, using constant values for ℓ_0, k and M. The initial apex height and velocity 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 ℓ_0, the maximum velocity $v_{act} = v_c + v_{nl}$ not to exceed ℓ_0/τ, 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. 7(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. 7(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 γ. As we can see, our



(a) (b)

Fig. 7. Subplot (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 subplot (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 γ: $\gamma \geq 10$ (black), $\gamma \geq 20$, $\gamma \geq 40$, $\gamma \geq 100$, and $\gamma \geq 200$ (light yellow).

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 (in terms of approximation error) of our proposed strategy for cancelling the nonlinear terms via our active SLIP control. We introduce the non-dimensional percentage errors of variables x, y and $\dot{x}$, respectively, as:

$$PE_x = 100\frac{\| x - \tilde{x} \|_2}{\ell_0}, \quad PE_y = 100\frac{\| y - \tilde{y} \|_2}{\ell_0}, \\ PE_{\dot{x}} = 100\frac{\| \dot{x} - \tilde{\dot{x}} \|_2}{\ell_0}\tau, \tag{13}$$

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 computed using Matlab numerical solver *ode45*, with absolute and relative tolerances set at 10^{-8}. The time constant τ has been chosen to be equal to $\tau = 1$ [s].

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 (13) for the approximation proposed in [19] versus our approximation with nonlinearity cancellation (4). Since the approximation in [19] 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. 8(a), 8(b) and 8(c) show the mean percentage errors PE_x, 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 we can see, our proposed strategy significantly reduces the percentage errors, especially for lower values of γ. This can serve as a starting point for the choice of γ while building a hardware prototype.

We now compute the percentage errors (13) for the actuated SLIP model, with our proposed actuation displacement strategy $\ell_{act}(t)$ as in (III-A). Fig. 9(a), 9(b) and 9(c) show the mean and standard deviation of, respectively, the percentage errors PE_x, PE_y and $PE_{\dot{x}}$, computed for several values of $\gamma \in [20,\ 200]$ and a set of $60{,}000$ initial conditions.

V. Control Actions: Online Computation of Optimal Parameters

We start by noting that the reachable $\{x,\ y,\ \dot{x}\}-$space obtained by regulating the actuator value (Fig. 5(b)), takes the form of a 2-dimensional surface. In order to increase the dimension of the reachable state-space, we need to add another degree of freedom: the touch-down angle θ_{TD}. Then, the reachable space becomes a non convex set in the 3-dimensional space, as shown in Fig. 10. In the formulation of our strategies, we will then consider three control parameters: the actuator values ℓ_{c1} and ℓ_{c2}, and the touch-down angle θ_{TD}.

The main advantage of using an approximation of the stance phase versus its numerical solution is given by the reduction

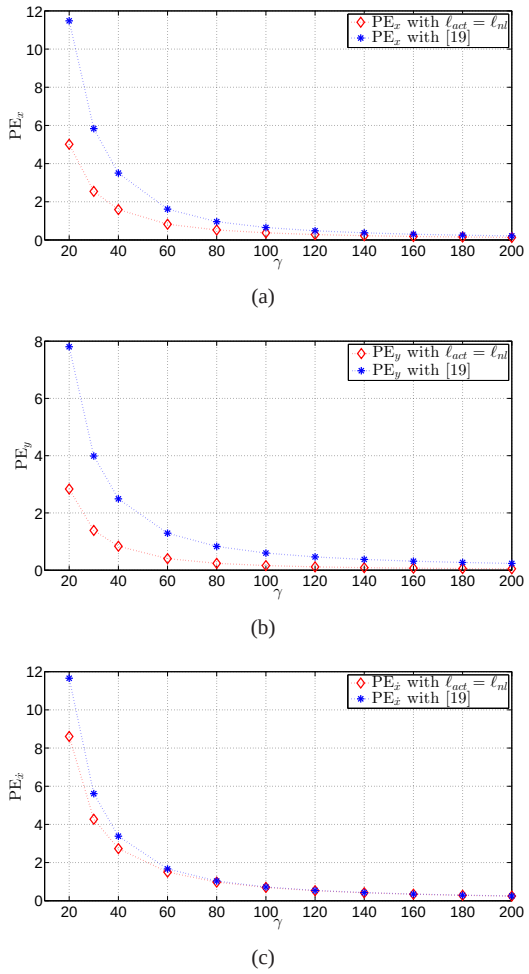


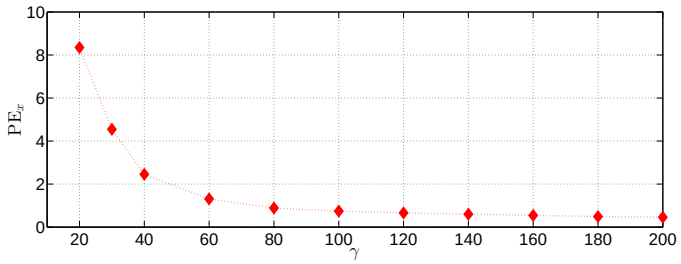
Fig. 8. The blue stars represent the PEs computed with respect to the stance phase approximation proposed in [19]. The red diamonds represent the PEs computed using our proposed approximation via nonlinearity cancellation. Initial apex conditions have been chosen to be $y \in [\ell_0, 2.5\ell_0]$, $\dot{x} \in [0.3\frac{\ell_0}{\tau},\ 3\frac{\ell_0}{\tau}]$, and $\theta_{TD} \in [85, 150]$ [deg]. Maximum actuator displacement and velocity are $max(|\ell_{act}|) = 0.05\ell_0$ and $max(|v_{act}|) = 0.5\ell_0/\tau$.

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 $9.4243 * 10^{-5}$ [s]: a decrease in computation time of over 250 times.

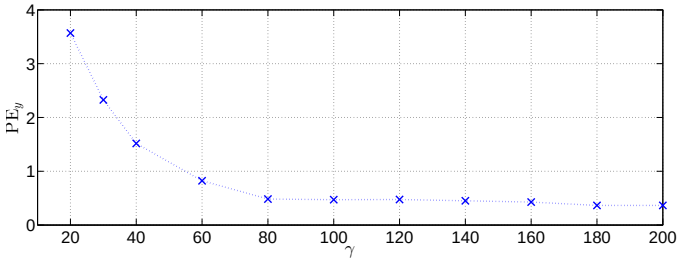
We start from this preliminary remark to introduce our proposed control actions.

A. Controlling the $\{y,\ \dot{x}\}-$state space

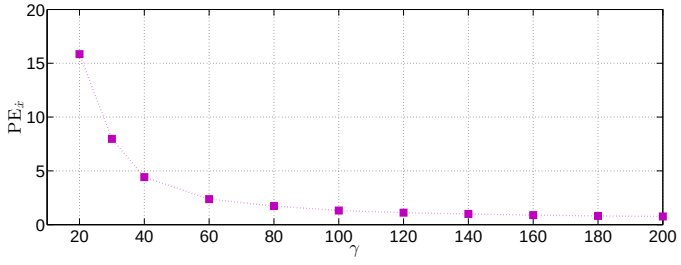
In this subsection we will focus our attention on controlling the height and velocity at apex only, disregarding the forward position. Therefore, the apex state becomes a two-dimensional vector $\boldsymbol{s} = \{y_a,\ \dot{x}_a\}$. We use a modified version of the Matlab function *fminsearch* (which optimizes constrained problems



(a)



(b)



(c)

Fig. 9. Percentage errors PE_x (a), PE_y (b) and $PE_{\dot{x}}$ (c) for our proposed approximation with actuation displacement (III-A). The colored symbols represent the mean values 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.

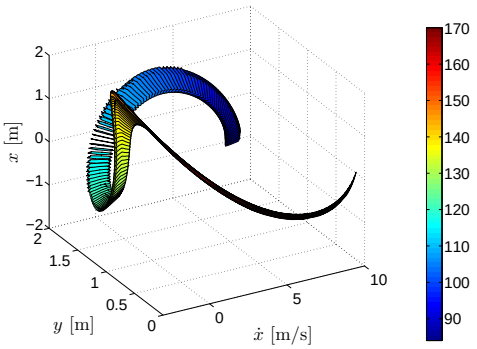


Fig. 10. Apex states $\{x, y, \dot{x}\}$ reachable in one step from the initial apex state $\{0, 1.4\ell_0, 2\frac{\ell_0}{\tau}\}$, for varying values of $\theta_{TD} \in [85, 170]$ [deg] (colorbar), ℓ_{c1} and $\ell_{c2} \in [-.05\ell_0, 0.5\ell_0]$. Actuator moves with velocity $v_c = 0.5\frac{\ell_0}{\tau}$. Relative spring stiffness $\gamma = 20$.

using the Nelder-Mead algorithm). At any current apex state $\boldsymbol{s}_n = \{y_n, \dot{x}_n\}$, we compute the values for the touch-down angle θ_{TD} and the two actuator values ℓ_{c1} and ℓ_{c2} that minimize in one step the distance to a desired apex $\boldsymbol{s}_{des} = \{y_{des}, \dot{x}_{des}\}$. 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.

At each step, n, the cost function to be minimized, $J(n)$, is defined as:

$$J(n) = 100\sqrt{\frac{(y_{n+1} - y_{des})^2}{\ell_0^2} + \frac{\tau^2(\dot{x}_{n+1} - \dot{x}_{des})^2}{\ell_0^2}}, \quad (14)$$

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

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 locomotion

Due to errors in the approximation (see Fig. 9(a), 9(b) and 9(c)), the touch-down angle and actuator values that minimize the cost function (14) may drive the system to an apex state that differs from the desired one by a certain amount. Therefore, the system will converge to an apex state that is not the desired one. In order to avoid this problem, we propose here a strategy to reduce such stead-state error, driving the system closer to the desired apex 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 $\{y_{des,0}, \dot{x}_{des,0}\}$. At each step n, we define the error between the actual and the approximate state as

$$\Delta y_n = y_n - y_{des,0}, \quad (15)$$
$$\Delta \dot{x}_n = \dot{x}_n - \dot{x}_{des,0}, \quad (16)$$

where y and $\dot{x}$ are the actual apex height and velocity of the system. At the $n-th$ step, we update the desired value for the next step $y_{des,n+1}$ and $\dot{x}_{des,n+1}$ to be

$$y_{des,n+1} = y_{des,n} - \sigma_1 \Delta y_n$$
$$\dot{x}_{des,n+1} = \dot{x}_{des,n} - \sigma_2 \Delta \dot{x}_n.$$

The proportional gains σ_1 and σ_2 are chosen to be $0 < \sigma_1 < 1$, $0 < \sigma_2 < 1$. The desired apex state is updated at each step, until the errors $\Delta y = 0$ and $\Delta \dot{x} = 0$, and the system reaches an equilibrium. Fig. 11(a), 11(b), 11(c) and 11(d) show, respectively, the percentage distance $J(n)$ from the desired apex state after $n = 1$, 3, 6 and 9 steps, on flat terrain, for $\gamma = 20$ and $\sigma_1 = \sigma_2 = 0.8$. Our proposed controller reduces the percentage distance J after 9 steps from a maximum of about 70% to a maximum of about 0.3%. Fig. shows an example of error reduction for 3 different initial conditions 12(a): as we can see, the error converges to zero.

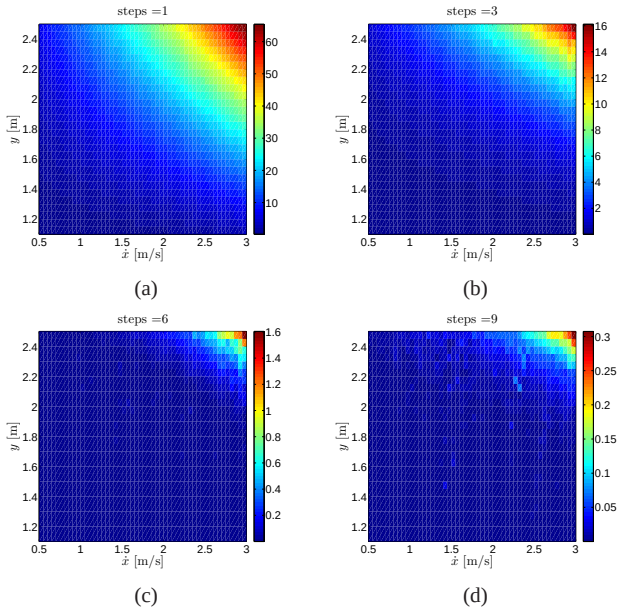


Fig. 11. These plots show the percentage error J after 1 (a), 3 (b), 6 (c) and 9 (d) steps, for $\gamma = 20$. The x-axis and y-axis represent the apex velocity and the apex height, respectively. Parameters σ_1 and σ_2 have been chosen to be equal to 0.8.

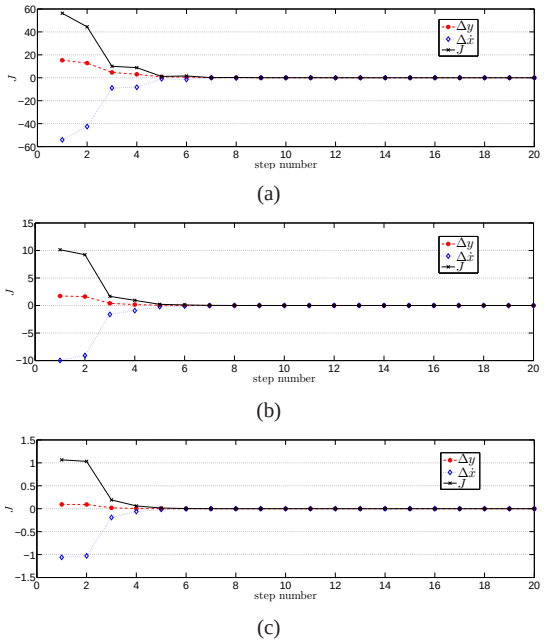


Fig. 12. Evolution at each step of the apex errors Δy (15) and $\Delta\dot{x}$ (16), and the cost function J (14) for inital and desired apex state $s = \{y_a,\, \dot{x}_a\} = \{2.4,\, 2.8\}$ (a), $\{1.8,\, 1.6\}$ (b), and $\{1.3,\, 0.8\}$ (c). As we can see, the errors converge to zero.

C. Foothold Placement Control

One of the key problems in legged locomotion on different kinds of terrain is to determine whether the foothold, i.e., the point at which the foot comes in contact with the terrain, is considered safe. For example, we can imagine the case of a terrain where only a specific set of N footholds, $x^f_{des,i}$, $i = \{1, \dots N\}$, is allowed, and everything else has to be avoided, as studied in [27]. We can then plan the trajectory to follow based on the knowledge we have of the terrain. In particular, at each step i we want to find a sequence of control actions θ_{TD}, ℓ_{c1}, and ℓ_{c2}, to minimize the distance between the desired footholds, $x^f_{des,i}$ and the actual landing of the foot, x^f:

$$d_i = \| \, x^f_i - x^f_{des,i} \, \|_2,$$

where the position of the foot at touch-down is computed as:

$$x^f = x_a + \dot{x}_a \sqrt{\frac{2}{g}(y_a + \ell_0 \cos\theta_{TD})} + \ell_0 \sin\theta_{TD}.$$

The foothold error is dependent on the planning *horizon*, i.e., the number of steps we can pre-compute. For a planning horizon of length N, we can write the following cost function to minimize:

$$J^f_N = \sum_{i=1}^{N} d_i^2.$$

However, to ensure that the hopper maintains a certain desired height, $y_{des,i}$, with respect to the terrain, we can modify the above cost to be:

$$J^f_N = \sum_{i=1}^{N} d_i^2 + w_y \sum_{i=1}^{N} (y_i - y_{des,i})^2,$$

where $w_y \in [0,\, 1]$ is the associated weight. Furthermore, to avoid that the optimal solution at the N-th step gives an unsuccessful jump, another element can be added to the cost function, and we obtain:

$$J^f_N = \sum_{i=1}^{N} d_i^2 + w \sum_{i=1}^{N} (y_i - y_{des,i})^2 + w_{\dot{x}} (\dot{x}_N - \dot{x}_{N-1})^2, \quad (17)$$

with $w_{\dot{x}} \in [0,\, 1]$.

Ideally, one would want an infinite planning horizon: $N = \infty$. In fact, the longer the horizon, the better the performance of the optimized problem (e.g., see Fig. 13). Though, the horizon length affects the computational time required to plan the desired trajectory online. This is where having an approximation for the stance phase becomes highly beneficial: in fact, being able to compute the optimal path via approximation significantly reduces the computational time, and as a consequence, it is possible to extend the planning horizon. However, one should keep in mind that the approximation, as such, carries an error: there is then a trade-off between horizon length/computation time, and foothold error. Fig. 14(a) and 14(b) show an example of trajectory planning on flat terrain, comparing a case in which the optimization problem has been solved using the approximate solution for the stance phase, and a case in which the numerical solution has been used. As expected, the minimization via numerical solution has a smaller error but a higher computational time compared to the minimization via approximation (on the same horizon length N). To be able to perform an online computation of the optimal control parameters, the computation time needs to be much smaller than the average time during flight: $t \ll 0.5$ [s]. Then, the only viable option for the numerical solution case is $N = 2$, and average foothold error over $10,000$ steps of 3.64 [cm]. For the approximate solution case, instead, one can use $N = 6$, and the mean foothold error is only 0.45 [cm], i.e., 8 times smaller than the numerical case.

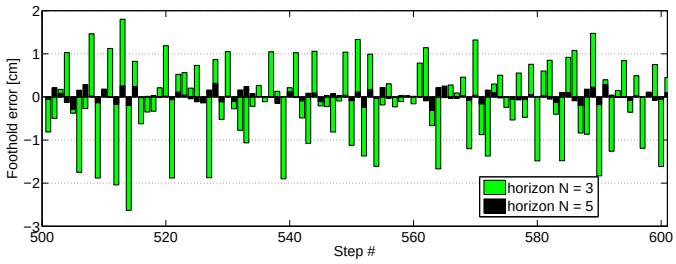
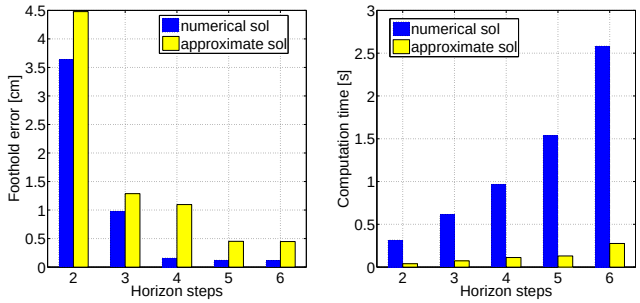


Fig. 13. Example of foothold error for 100 steps for the case with horizon $N = 3$ (green bars) and $N = 5$ (black bars). As expected, a higher planning horizon gives better performances in terms of error.



(a) (b)

Fig. 14. These plots show the mean foothold error (a) and the mean computation time (b) over 10,000 steps, for trajectory planning, for different horizon steps. The terrain has been chosen to be flat, with footholds drawn from a uniform distribution on the open interval $(0.3\ell_0,\ \ell_0)$. The blue dotted bars refer to the optimization using the numerically computed solution, while the yellow bars refer to the optimization computed using the approximate solution.

VI. Applications

In this section we show the performance of our controller to perturbations in terrain. In particular, we test recovery for 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 data for a typical human. In [3], the relative spring stiffness γ was found to be very similar for different gaits, such as running and hopping, of various animals. In particular, values were computed to be between 7.1 and 14.6 for runners, and 7.7 and 13.6 for hoppers. Because of the limitations shown in Fig. 7(a) and 7(b), for our simulations we use $\gamma = 20$, which is slightly higher than the average biological value, but it 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 ℓ_{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^2
$M =$	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 50% of the leg length ℓ_0.

At each apex state, we use the strategy in V-A to compute the optimal values for ℓ_{c1}, ℓ_{c2} and θ_{TD} for flat terrain. Once the leg touches the ground with the computed touch-down angle and the desired ℓ_{c1}, the strategy in V-A is simulated again during the first half of the stance phase, this time to compute only ℓ_{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. 15(a), 15(b), 15(c), and 15(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 8 steps. Fig. 16(a) and 16(b) show an example of apex recovery for an unforeseen drop of magnitude 50% 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 estimates of upcoming terrain height 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 Fig. 17(a), 17(b), and 17(c), the system is able to successfully hop on a rough terrain with a maximum magnitude landing height of 0.26 [m] and a maximum perturbation of 0.46 [m], i.e., 46% of the leg length. Consistent with what is in Fig. 9(b) and 9(c), the error on $\dot{x}$ is on average higher than the error on y, and in this example they both do not exceed 4%. Note that, since the terrain height varies continuously, the energy of the system varies at each step.

The strategy of recomputing ℓ_{c2} during the first stance of the stance phase after encountering a perturbation, can dramatically improve the performances of our controller. Indeed, path on the same terrain has been studied using the computation during flight only, and the recomputing of ℓ_{c2} during the first half of stance. As we can see from Fig. 17(c), the cost function

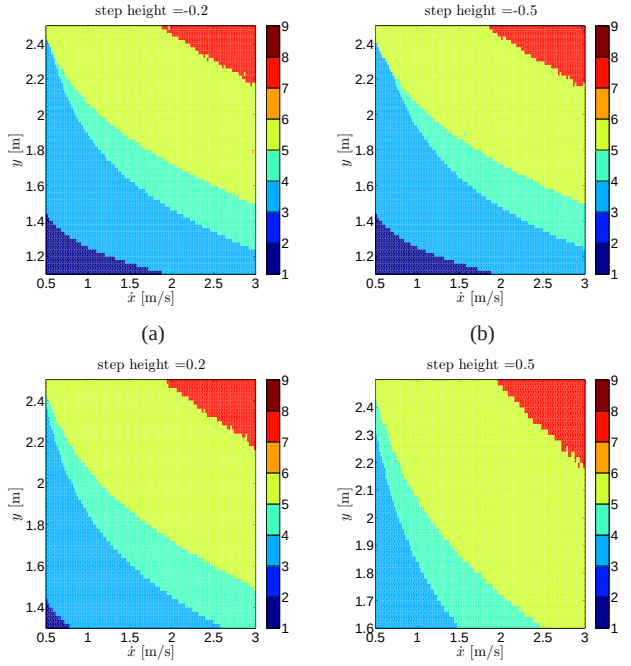


Fig. 15. Number of steps (colorbar) to reach 1% of desired value, for terrain perturbation of magnitude (a) -0.2 [m], (b) -0.5 [m], (c) 0.2 [m], (d) 0.5 [m]. Simulation parameters are chosen as per Table I. Note that in case of a positive terrain perturbation (subplots (c) and (d)), the apex height y has been chosen to leave enough room to the leg to swing during flight without colliding with the terrain.

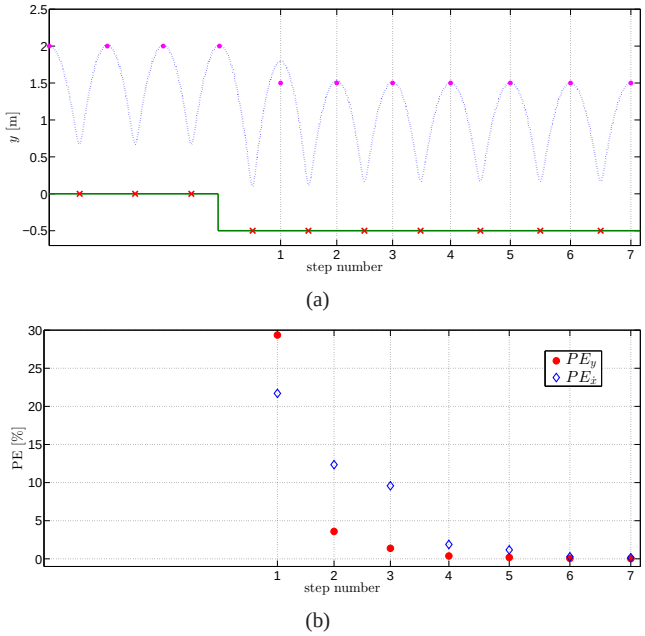


Fig. 16. Plot (a) shows the trajectory of the active SLIP for initial apex state $y = 2$ [m], $\dot{x} = 1.3$ [m/s], and negative terrain perturbation of magnitude 0.6 [m]. The blue dotted line represent the trajectory of the mass, the purple circles the desired apex height. The green solid line is the terrain height and the red crosses are the landing points of the foot. Plot (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.

J (computed as in (14)) is significantly smaller than the case without controller update.

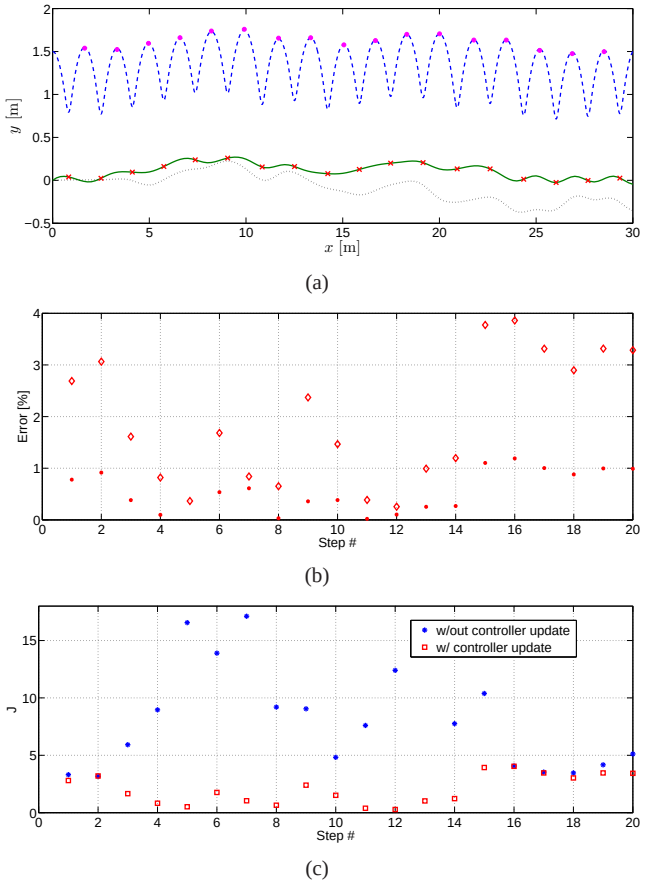
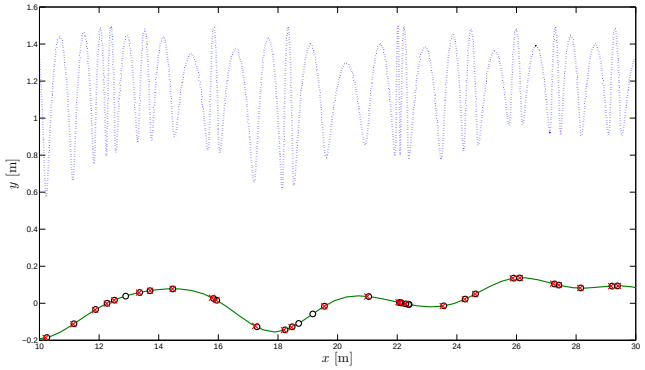


Fig. 17. Plot (a) shows the trajectory of the active SLIP hopping on a random-generated rough terrain. The controller acts to maintain a constant apex height with respect to the terrain of $y = 1.5$ [m], and a constant forward velocity of $\dot{x} = 2$ [m/s]. The blue dotted line represent the trajectory 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. Plot (b) shows the percentage error of apex height PE_y (circle) and velocity $PE_{\dot{x}}$ (diamond) at each step. Plot (c) shows the cost J as in (14) for the case without (blue star) and with (red square) controller update during the first part of the stance phase.

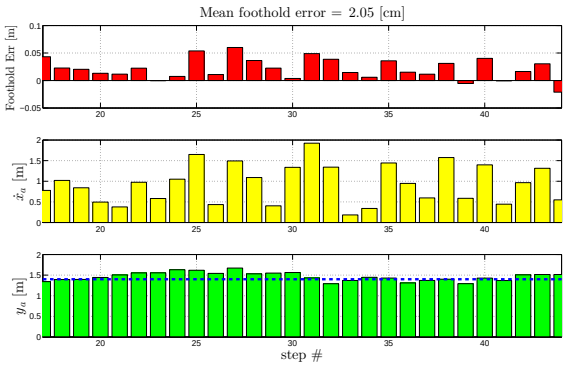
C. Landing on feasible footholds

We show here an example of hopping on a set of predefined footholds.

The set of footholds on an uneven terrain have been generated taking points from a uniform distribution between 0.6 [m] and 1.2 [m]. The cost function the be minimized is defined in Eq. (17), with weights $w_y = 0.8$ and $w_{\dot{x}} = 1$. We used a least-square algorithm to find the optimal solution for a planning horizon of $N = 4$ steps, for a total of 50 steps. Note that, according to the specifics of our minimization algorithm, the hopper was allowed to skip some footholds if pertinent. Fig. 18(a) shows a close-up of the output trajectory, while Fig. 18(b) shows at each step the foothold error (i.e., the distance from the desired to the actual foothold), the forward velocity and the apex height. The desired apex height with respect to the terrain was set as $y_{des} = 1.4$ [m]. As we can see, over 50 steps, the mean foothold error was 2.05 [cm].



(a)



(b)

Fig. 18. Plot (a) shows the trajectory of the point mass (blue dotted line) on a terrain (green line). The white circles are the desired footholds, while the red crosses are the actual placements of the foot. As noted in the text, our minimization algorithm allows for some footholds to be skipped. Plot (b) from top to bottom shows the foothold error (red bars), the forward velocity (yellow bars) and the apex height with respect to the terrain height (green bars). The dotted blue line represents the desired apex height.

VII. Conclusions

In this work, we consider the active SLIP model, a non-conservative extension of the classic SLIP that adds an actuator in series with the spring. We propose 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 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 or a desired set of terrain footholds, and we validate it through simulations, even in the presence of perturbations on the terrain height. Furthermore, we provide a feasibility study of our strategy for different system parameters.

Acknowledgments

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

References

[1] G. Piovan and K. Byl, "Two-element control for the active SLIP model," in *IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, Germany, May 2013, pp. 5636–5642.

[2] R. Blickhan, "The spring-mass model for running and hopping," *Journal of Biomechanics*, vol. 22, no. 11/12, pp. 1217–1227, 1989.

[3] 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.

[4] 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.

[5] M. H. Raibert, *Legged Robots that Balance*. MIT Press, 1986.

[6] G. Zeglin, "The Bow Leg hopping robot," Ph.D. dissertation, Carnegie Mellon University, Pittburgh, PA, USA, Oct. 1999.

[7] U. Saranli, M. Buehler, and D. E. Koditschek, "RHex: A simple and highly mobile hexapod robot," *International Journal of Robotics Research*, vol. 20, pp. 616–631, 2001.

[8] R. Altendorfer, U. Saranli, H. Komsuoglu, D. Koditschek, H. B. Brown, M. Buehler, N. Moore, D. McMordie, and R. Full, "Evidence for spring loaded inverted pendulum running in a hexapod robot," in *Experimental Robotics VII*, D. Rus and S. Singh, Eds. Springer-Verlag, 2001, pp. 291 – 302.

[9] 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.

[10] R. M. Ghigliazza, R. Altendorfer, P. Holmes, and D. Koditschek, "A simply stabilized running model," *SIAM Rev.*, vol. 47, no. 3, Mar. 2005.

[11] S. Riese and A. Seyfarth, "Stance leg control: variation of leg parameters supports stable hopping," *Bioinspiration & Biomimetics*, vol. 7, no. 1, p. 016006, 2012.

[12] 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.

[13] B. Andrews, B. Miller, J. Schmitt, and J. E. Clark, "Running over unknown rough terrain with a one-legged planar robot," *Bioinspiration & Biomimetics*, vol. 6, no. 2, p. 026009, 2011.

[14] 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.

[15] A. Seyfarth, H. Geyer, and H. Herr, "Swing-leg retraction: a simple control model for stable running," *Journal of Experimental Biology*, vol. 206, pp. 2547–2555, 2003.

[16] J. G. D. Karssen, M. Haberland, M. Wisse, and S. Kim, "The optimal swing-leg retraction rate for running," in *IEEE Int. Conf. on Robotics and Automation*, Shangai, China, May 2011, pp. 4000–4006.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] D. Dudek and R. J. Full, "Passive mechanical properties of legs from running insects," *Journal of Experimental Biology*, vol. 209, pp. 1502–1515, 2006.

[22] 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.

[23] 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.

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

[25] G. Piovan 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.

[26] K. Byl, M. Byl, M. Rutschmann, B. Satzinger, L. van Blarigan, G. Piovan, and J. Cortell, "Series-elastic actuation prototype for rough terrain hopping," in *IEEE International Conference on Technologies for Practical Robot Applications*, 2012, pp. 103–110.

[27] M. Rutschmann, B. Satzinger, M. Byl, and K. Byl, "Nonlinear model predictive control for rough terrain hopping," in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, Villamoura, Algarve, Portugal, Oct. 2012, pp. 1859–1864.

[28] 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.