

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

Giulia Piovan and Katie Byl

Abstract—The Spring Loaded Inverted Pendulum (SLIP) has been extensively studied and used as a model capturing general aspects 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 added series actuation. In particular, we propose a partial feedback linearization action for actuator displacement to analytically solve part of its dynamics, thereby reducing computational time and increasing the practicality of performing online control actions. This is then paired with a two-part control action to add/remove energy to/from the system and modify the upcoming apex state to span an open set within the reachable apex states. 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 computation errors by the use of an approximation of the leg-angle dynamics during the stance phase, and we demonstrate the proposed strategy on a more dynamically sophisticated planar hopper model.

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.

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.

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], and [8]). Consequently, various control strategies have been proposed. Some studies consider stable periodic gaits: e.g. [9] prove that setting a specific fixed touch-down angle produces asymptotically stable periodic gaits, and [10] investigates stability of vertical hopping while varying spring stiffness and leg length. Other studies focus on negotiating uneven terrains: e.g., [11] propose three strategies for step-length adjustment that involve changing either one of forward speed, stance time or flight time, while keeping the other two constant, [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 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. It is clearly important to examine the effect of adding actuation 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. Theoretical work in [28], integrated by experimental work in [29], considers a system with a series elastic actuator at the leg, and a rotational actuator and a spring at the hip. The control action proposed exploits the system's passive dynamics to minimize energy consumption and reach a steady state gait at a target forward speed.

This work adds several contributions. First, we consider the active SLIP model and propose a strategy for actuator displacement. On one hand, it allows us to analytically solve the equation that describes the leg-length dynamics during stance via partial feedback linearization. On the other, by dividing the stance phase in two parts defined by the point of maximum leg compression and setting two different actuation values for each part, it allows us to add or remove energy and to modify the upcoming apex state to span an open set within the reachable apex state set. Second, we provide a feasibility study of our strategy for different system parameters. 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 test our strategy to deal with terrain perturbations, and, for a set of system parameters, we quantitatively define the number of jumps necessary for full recovery. We additionally extend the proposed approximation to the case of a more realistic hopper model with leg mass and body inertia.

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, which is employed to analytically solve part of the system's dynamics and to add/remove energy to/from the system. 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. It also contains an extension of our proposed approximation strategy to a more complex planar hopper model, treated in more detail in Appendix. 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 an 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 , as shown in Fig. 1. Dynamics for the SLIP model occur on the sagittal plane, and constitute a 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:

$$\begin{aligned}\ddot{x}(t) &= 0, \\ \ddot{y}(t) &= -g,\end{aligned}$$

where, as customary, g is 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 $\mathbf{s} = \{x_a, y_a, \dot{x}_a\}$.

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 as follows:

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

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

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

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

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

In this work, we call a *jump* the transition from one apex state to the next. A jump is *successful* if the forward velocity during the entire trajectory takes only 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 larger than the leg length at equilibrium, ℓ_0 .

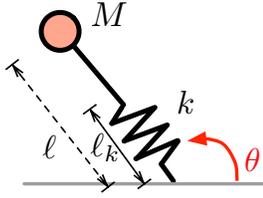


Fig. 1. Classic SLIP model (passive SLIP)

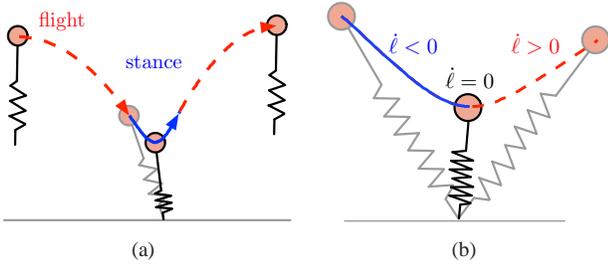


Fig. 2. (a) Scheme that illustrates the phases of the SLIP model trajectory, and (b) stance phase

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 [30], 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

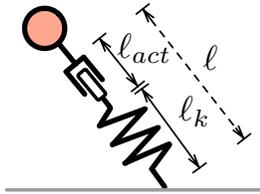


Fig. 3. Active SLIP model (active SLIP)

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

A. Approximating the stance phase dynamics through partial feedback linearization

We propose a strategy for the actuator displacement $\ell_{act}(t)$ as follows. Let us write the total actuator displacement as: $\ell_{act}(t) = \ell_{nl}(t) + \ell_c(t)$, 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):

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

Note that, at touch-down, this becomes:

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

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 particular 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, $\ell_c(t_i)$, until it reaches the desired value $\bar{\ell}_c$:

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

as illustrated in Fig. 4. Note: in general, $v_{nl}(t)$ is not a constant

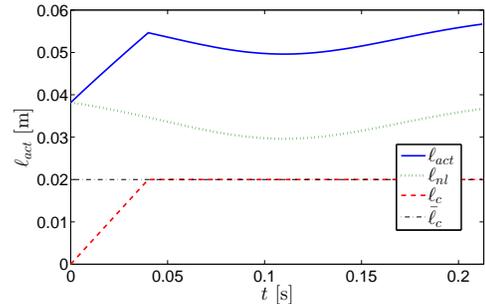


Fig. 4. Example of the total actuator value $\ell_{act}(t)$ during stance phase, and its two components $\ell_{nl}(t)$ and $\ell_c(t)$.

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

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

where¹

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

$$a = \frac{\ell(t_i) - \ell_0 - \ell_c(t_i)}{\cos \beta}, \quad r = \ell_0 + \ell_c(t_i),$$

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

$$\beta = \text{atan}_2\left(-\frac{\dot{\ell}(t_c)}{\omega}, \ell(t_c) - \ell_0 - \bar{\ell}_c\right),$$

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

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 dynamics of the leg length during stance are described by Equation (6). Let us define $\alpha(t) = \theta(t) - \pi/2$, and assume that both $\alpha(t)$ and the angular span $\Delta\alpha = \alpha_{TD} - \alpha_{TO}$ are sufficiently small to be simplified as $\sin(\alpha) \approx \alpha$. Then, equation (2) can be simplified and re-written in terms of α , becoming:

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

Toward solving for a corresponding analytic approximation for $\alpha(t)$, we introduce the variables $u(t)$ and $p(t)$ such as $\alpha(t) = p(t)u(t)$. Substituting $\alpha(t)$ in (7), we 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

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

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

$$\ddot{u} - u\left(\frac{\ddot{\ell} + g}{\ell}\right) = 0. \quad (8)$$

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

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

¹For any point (x, y) in the xy -plane minus 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.

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

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

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

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

and thereby obtain the Mathieu equation

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

with

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

Equation (9) can be developed as

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

We then obtain

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

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:

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

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

$$-1 + \frac{1 + \zeta + \xi\psi}{1 + \kappa \cos \psi + \xi\psi} \simeq$$

$$\tilde{\lambda}^2 - \tilde{\delta}(\tilde{\epsilon} \cos \psi - \tilde{\epsilon}^2 \cos^2 \psi + \tilde{\epsilon}^3 \cos^3 \psi - \dots),$$

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

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

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

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

B. Choice of the actuator constant value ℓ_c : two-element strategy

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. 5.

Dissimilarly from Raibert's famous single-legged hopper, which adjusts the energy via actuation once at mid-stance, we propose to adjust the actuator motion twice during stance. This approach allows us not only to regulate the system's energy, but, more particularly, its two components of forward velocity and apex height. As Fig. 5 illustrates, actuation during the entire stance phase allows the system to reach a wider variety of apex states than would otherwise be reached with mid-stance actuation only. Specifically, using a single (1-dimension) thrust parameterization will clearly map to only a 1-dimensional set of apex states, while the full reachable set of apex states is an approximately 2-dimensional surface ([25], [27]).

Furthermore, the decision of limiting our control to a two-parameter (ℓ_{c1} , ℓ_{c2}) choice 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. 6(a) and 6(b) provide an example of how, by setting only two actuator values, it is possible to reach in one jump a wide range of apex states, influencing both apex height, apex velocity and apex forward position, in all reachable directions. Because the reachable $\{x, y, \dot{x}\}$ -space obtained by regulating the actuator value (Fig. 6(b)) takes the form of a 2-dimensional surface, increasing the dimension of the reachable state-space requires that we add another degree of freedom. This can be done by varying the touch-down angle θ_{TD} . Then, the reachable space becomes a non-convex set in the 3-dimensional space, as shown in Fig. 7.

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

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

Fig. 8 shows the change in energy from one initial apex state to the next for different values for the couple (ℓ_{c1}, ℓ_{c2}) . As

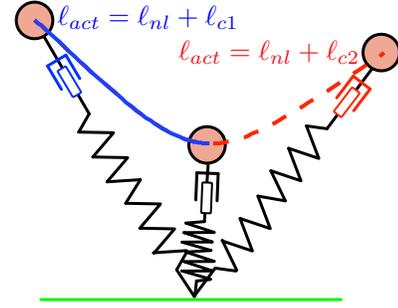


Fig. 5. Actuator motion during stance phase

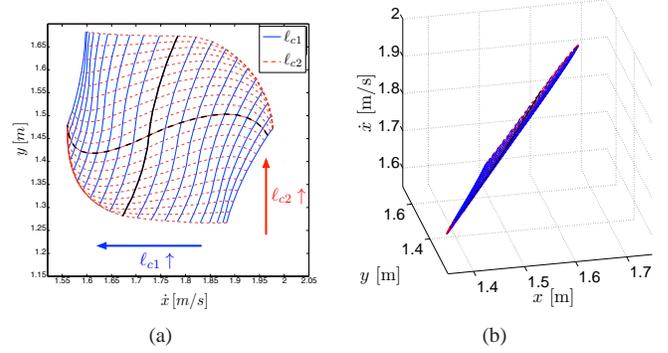


Fig. 6. Apex states $\{y, \dot{x}\}$ (a), and $\{x, y, \dot{x}\}$ (b), reachable in one jump, with $M = 10$ [kg], $k = 1962$ [N/m], $\ell_0 = 1$ [m]. Initial apex: $x_{apex} = 0$ [m], $y_{apex} = 1.4$ [m], $\dot{x}_{apex} = 2$ [m/s], $\theta_{TD} = 103$ [deg]. Maximum/minimum actuator length: ± 0.05 [m]. Actuator moves with velocity $v_c = 0.5$ [m/s]. Each solid blue line corresponds to a different actuator value during the first half of the stance phase, ℓ_{c1} , while each dotted red line corresponds to a different actuator value during the second half of the stance phase, ℓ_{c2} . The black lines correspond to the case of either $\ell_{c1} = 0$ or $\ell_{c2} = 0$.

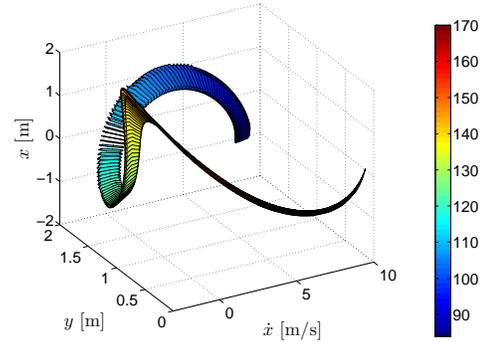


Fig. 7. Apex states $\{x, y, \dot{x}\}$ reachable by our control strategy in one jump from the initial apex state $\{0, 1.4\ell_0, 2\frac{\ell_0}{\tau}\}$, for varying values of $\theta_{TD} \in [85, 170]$ [deg] (colorbar), ℓ_{c1} and $\ell_{c2} \in [-.05\ell_0, 0.5\ell_0]$. Actuator moves with velocity $v_c = 0.5\frac{\ell_0}{\tau}$. Relative spring stiffness $\gamma = 20$.

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.

The lack of body inertia in this simplified model prevents

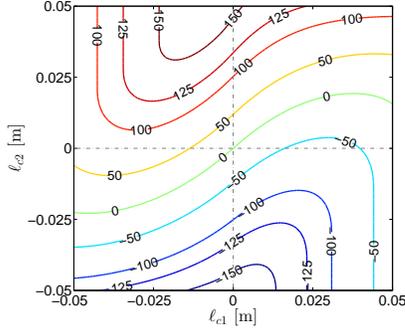


Fig. 8. The contour plot represents the energy change values $\Delta E = E_{a,2} - E_{a,1}$. $E_{a,1}$ is the energy (11) at the initial apex state $y_a = 1.4$ [m] and $\dot{x}_a = 2$ [m/s]. $E_{a,2}$ is the energy (11) at each next apex state computed for $\theta_{TD} = 103$ [deg] and varying values of the couple (ℓ_{c1}, ℓ_{c2}) .

the use of body attitude as additional control action to reach a desired state. However, it is not unreasonable to believe that the strategy we propose can be easily paired with other leg placement and body attitude control strategies. For example, the leg actuator proportional controller proposed in [28] could be replaced by a two-part thrust actuation strategy such as ours, preserving the energy efficient hip actuator controller based on the hip passive oscillations.

IV. PERFORMANCE

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

$$\gamma = \frac{k\ell_0}{Mg}. \quad (12)$$

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 and M , and varying k . 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. To include such feasibility constraints in our work, we assume the total actuator displacement $\ell_{act}(t) = \ell_c(t) + \ell_{nl}(t)$ must not exceed 10% of the leg length ℓ_0 , the maximum velocity $v_{act} = v_c + v_{nl}$ not exceed ℓ_0/τ , and that $\ell_{act}(t)$ at any given time must not bottom-out the spring, i.e., $\ell_k(t) \geq \ell_{k,min}$. The allowable amount of displacement and velocity for the nonlinear part (ℓ_{nl}) and the constant part (ℓ_c) can be allocated in an infinite number of ways. For example, Fig. 9(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. 9(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 . Our strategy works

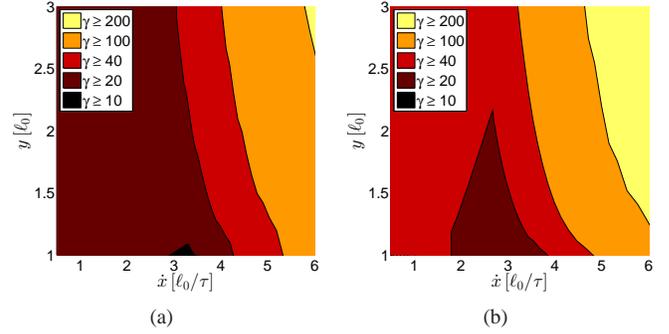


Fig. 9. Subplot (a) shows the initial apex states that require $\ell_{nl} \leq 0.5\ell_0$ and $v_{nl} \leq 0.5\ell_0/\tau$ to perform a symmetric jump, while subplot (b) considers apex states that require $\ell_{nl} \leq 0.4\ell_0$ and $v_{nl} \leq 0.3\ell_0/\tau$. Different shades refer to different intervals of γ : $\gamma \geq 10$ (black), $\gamma \geq 20$, $\gamma \geq 40$, $\gamma \geq 100$, and $\gamma \geq 200$ (light yellow).

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:

$$PE_x = 100 \frac{\|x - \tilde{x}\|_2}{\ell_0}, \quad PE_y = 100 \frac{\|y - \tilde{y}\|_2}{\ell_0}, \quad (13)$$

$$PE_{\dot{x}} = 100 \frac{\|\dot{x} - \tilde{\dot{x}}\|_2}{\ell_0} \tau,$$

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. 10(a), 10(b) and 10(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). We can see that 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.

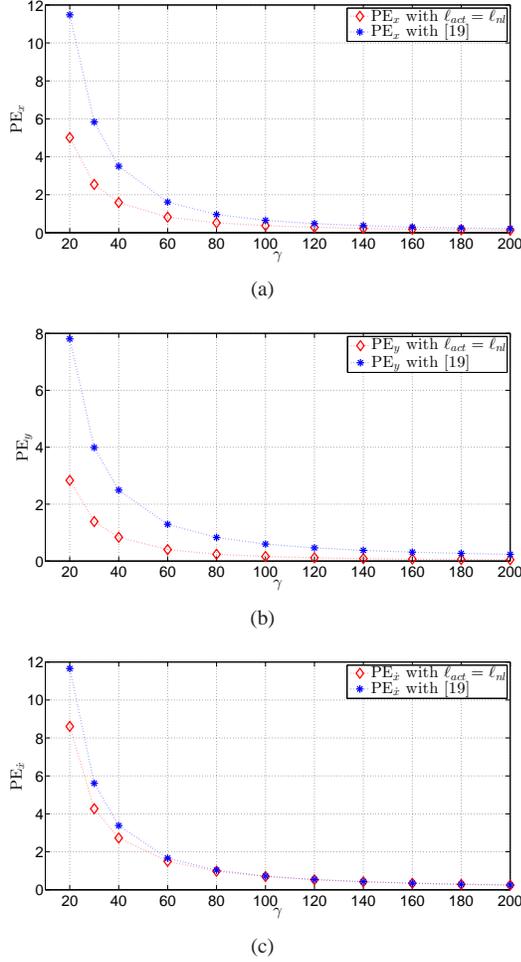


Fig. 10. The blue stars represent the PEs computed with respect to the stance phase approximation proposed in [19]. The red diamonds represent the PEs computed using our proposed approximation via nonlinearity cancellation. Initial apex conditions have been chosen to be $y \in [\ell_0, 2.5\ell_0]$, $\dot{x} \in [0.3\frac{\ell_0}{\tau}, 3\frac{\ell_0}{\tau}]$, and $\theta_{TD} \in [85, 150]$ [deg]. Maximum actuator displacement and velocity are $\max(|\ell_{act}|) = 0.05\ell_0$ and $\max(|v_{act}|) = 0.5\ell_0/\tau$.

We now compute the percentage errors (13) for the actuated SLIP model, with our proposed actuator displacement strategy $\ell_{act}(t)$ as in (III-A). Fig. 11(a), 11(b) and 11(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.

C. Robustness to sensor noise

The exact solution of the leg length dynamics via partial feedback linearization and, more generally, the approximation of the overall stance dynamics, depend on the accuracy of the sensors' measurements available. We test the percentage errors (13) for the actuated SLIP model when sensor noise is present. In particular, we consider errors in the detection of the initial apex height, position and forward velocity, which translates into errors of the stance phase's initial conditions, and noise on the partial feedback linearization term (4). Fig. 12(a), 12(b), and 12(c) show the mean percentage errors when sensor noise is present, and compare it with the noiseless

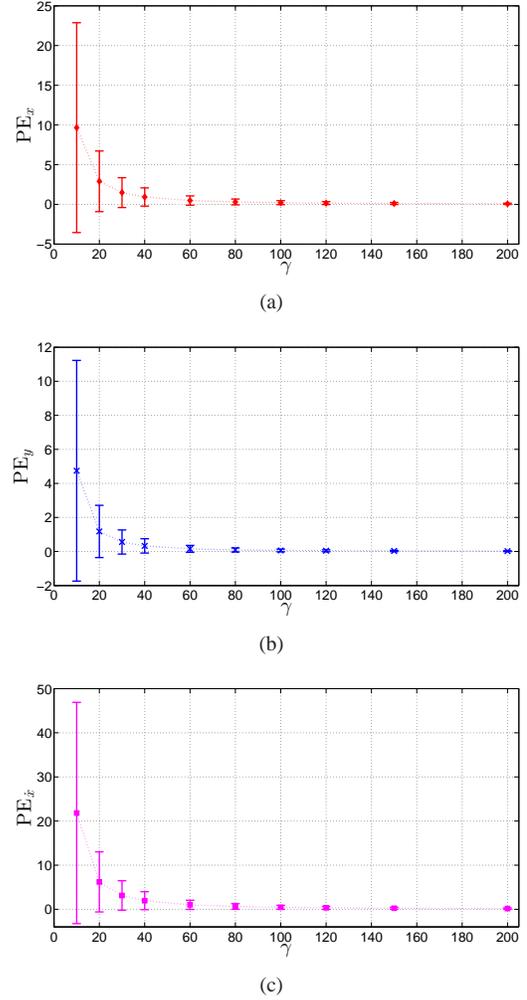


Fig. 11. Percentage errors PE_x (a), PE_y (b) and $PE_{\dot{x}}$ (c) for our proposed approximation with actuator displacement (III-A). The colored symbols represent the mean values of the percentage errors, and the vertical bars the respective standard deviations, computed for a pool of 60,000 initial conditions. The initial apex conditions have been chosen to be $y \in [\ell_0, 1.8\ell_0]$, $\dot{x} \in [0.5\frac{\ell_0}{\tau}, 3\frac{\ell_0}{\tau}]$, and $\theta_{TD} \in [90, 135]$ [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.

case. Sensor noise on the initial apex state is taken from a uniform distribution within $\pm 1\%$ the actual values. Furthermore, we assume the controller is updated at 1000 Hz, and the partial feedback linearization term (4) is affected by measurement noise modelled as gaussian distribution with zero mean and standard deviations $\sigma = 0.1$ for θ and $\dot{\theta}$, and $\sigma = 0.01$ for ℓ . While the presence of noise degrades the accuracy of the approximation of the system's dynamics, the noise considered still permits to obtain a good level of accuracy.

D. Cost of Transport

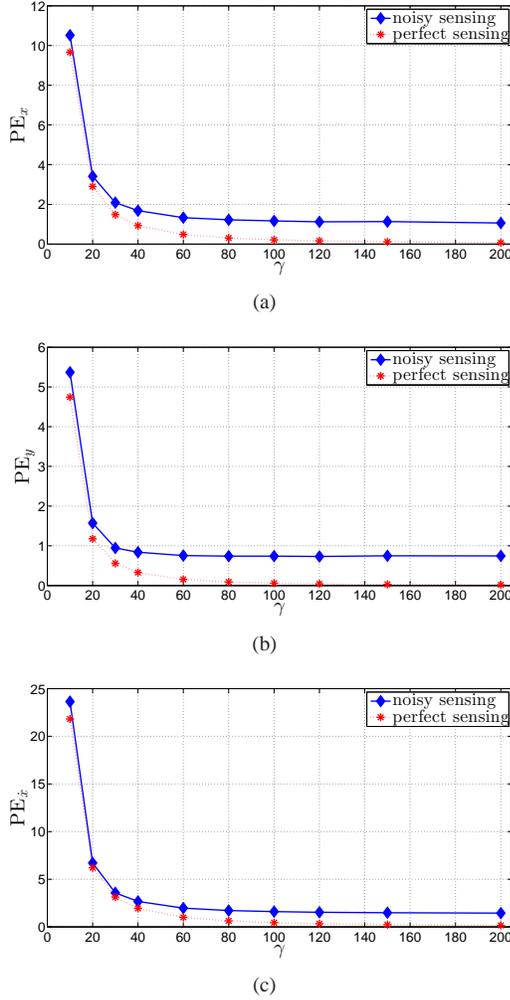


Fig. 12. Percentage errors PE_x (a), PE_y (b) and PE_x (c) for our proposed approximation with actuator displacement (III-A), with and without sensor noise. The blue diamond dotted lines represent the mean values of the percentage errors with noisy sensors, while the red star solid lines represent the mean values of the percentage errors with perfect sensing, for varying values of γ . PEs are computed for a pool of 60,000 initial conditions, chosen to be $y \in [l_0, 1.8l_0]$, $\dot{x} \in [0.5\frac{l_0}{\tau}, 3\frac{l_0}{\tau}]$, and $\theta_{TD} \in [90, 135]$ [deg]. Maximum actuator displacement and velocity are $\max(|\ell_{act}|) = 0.1l_0$ and $\max(|v_{act}|) = l_0/\tau$, with $\ell_c \in [-0.05l_0, 0.05l_0]$, and $v_c \in [0.5l_0/\tau, 0.5l_0/\tau]$. Sensor noise is modelled as follows. Noise on the initial apex states are taken from a uniform distribution within $\pm 1\%$ of the actual values, while the partial feedback linearization term (4) noise is modelled as gaussian distribution with zero mean and standard deviations $\sigma = 0.1$ for θ and $\dot{\theta}$, and $\sigma = 0.01$ for ℓ .

At each time t during the stance phase, it is possible to compute the total work done by the spring as:

$$W_{spring}(t) = \int_{t_{TD}}^t k(\ell_k(t) - \ell_{k,0})\dot{\ell}(t)dt. \quad (14)$$

Note that $\dot{\ell}(t) = \dot{\ell}_k(t) + \dot{\ell}_{act}(t)$, where $\dot{\ell}_{act}(t)$ is the actuator's velocity.

The energy stored in the spring is

$$E_k(t) = \frac{k}{2}(\ell_k(t) - \ell_{k,0})^2.$$

If there is no actuation, the energy stored in the spring is equivalent to the work done by the spring, i.e., $E_k(t) = W_{spring}(t)$.

However, when actuation is present, it is responsible for part of the energy stored in the spring: $E_k(t) = W_{spring}(t) + W_{act}(t)$, where $W_{act}(t)$ is the work done by the actuator. Then

$$\begin{aligned} W_{act}(t) &= E_k(t) - W_{spring}(t) \\ &= \frac{k}{2}(\ell_k(t) - \ell_{k,0})^2 - \int_{t_{TD}}^t k(\ell_k(t) - \ell_{k,0})\dot{\ell}(t)dt \\ &= \dots \\ &= \int_{t_{TD}}^t P_{act}(t)dt + \frac{k}{2}(\ell_k(t_{TD}) - \ell_{k,0})^2, \end{aligned}$$

and $P_{act}(t) = -k(\ell_k(t) - \ell_{k,0})\dot{\ell}_{act}(t)$ is the power required by the actuator.

The dimensionless specific cost of transport is defined as:

$$\cot \triangleq \frac{W_{act,tot}}{Mgd}, \quad (15)$$

where d is the total distance travelled by the system, and $W_{act,tot}$ is the total work done but the actuator for the entire stance phase. To avoid power regeneration, it is appropriate to compute $W_{act,tot}$ as the integral of the unsigned power:

$$W_{act,tot} = \int_{t_{TD}}^{t_{TO}} |P_{act}(t)|dt + \frac{k}{2}(\ell_k(t_0) - \ell_{k,0})^2.$$

In our case, the total cost of transport can be computed for one jump (apex-to-apex), and therefore the distance travelled d is equal to the distance travelled from one initial apex state to the next. The work done by the actuator is a combination of the work done to cancel the nonlinearity as in (4), and the work required to move the actuator at the desired constant values $\ell_{c,1}$ and $\ell_{c,2}$. Fig. 13(a) shows the cost of transport computed for symmetric jumps starting at different initial conditions, where the only actuation considered is the one due to ℓ_{nl} , while $\ell_{c,1}$ and $\ell_{c,2}$ are set to zero. For varying initial conditions, cancelling the non-linear terms in equation implies a cost of transport $\cot < 0.21$. One should note that the considered jumps are, in general, passively non-symmetric, and therefore the cost of transport computed includes the cost to drive many non-symmetric jumps to be symmetric. Fig. 13(b) shows instead the cost of transport for an assigned initial conditions, considering both the effect of ℓ_{nl} and of varying values of ℓ_{c1} and ℓ_{c2} . Also in this case the cost of transport is $\cot < 0.26$.

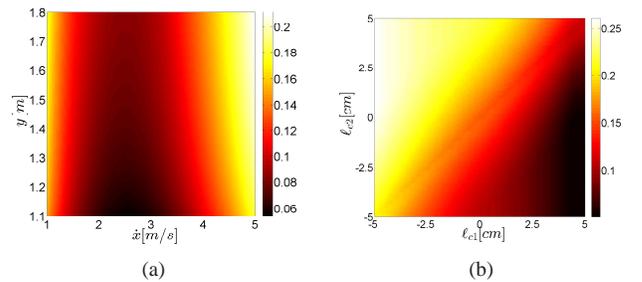


Fig. 13. Fig. 13(a) illustrates the cost of transport computed for symmetric jumps starting at different initial conditions, where the only actuation considered is the one due to ℓ_{nl} , while $\ell_{c,1}$ and $\ell_{c,2}$ are set to zero. Fig. 13(b) shows instead the cost of transport for an assigned initial apex state, $y_a = 1.3$ [m], $\dot{x}_a = 2$ [m/s], and $\theta_{TD} = 78$ [deg]. Both the effect of ℓ_{nl} and of varying values of ℓ_{c1} and ℓ_{c2} are considered.

V. CONTROL ACTIONS: ONLINE COMPUTATION OF OPTIMAL PARAMETERS

We start by noting that, In order to increase the dimension of the reachable state-space to a 3-dimensional space, we need to add another degree of freedom: the touch-down angle θ_{TD} , as illustrated in Fig. 7. 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 a reduction of computational time, which increases the practicality 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 $s = \{y_a, \dot{x}_a\}$. We use a modified version of the Matlab function *fminsearch* (which optimizes constrained problems using the Nelder-Mead algorithm). At any current apex state $s_n = \{y_n, \dot{x}_n\}$, 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 jump the distance to a desired apex $s_{des} = \{y_{des}, \dot{x}_{des}\}$. 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:

$$J(n) = 100 \sqrt{\frac{(y_{n+1} - y_{des})^2}{\ell_0^2} + \frac{\tau^2(\dot{x}_{n+1} - \dot{x}_{des})^2}{\ell_0^2}}, \quad (16)$$

which expresses the percentage distance from the next apex state $s_{n+1} = \{y_{n+1}, \dot{x}_{n+1}\}$ to the desired one, 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. 11(a), 11(b) and 11(c)), the touch-down angle and actuator values that

minimize the cost function in Equation (16) 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 address this problem, we propose here a strategy to reduce such steady-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

$$\Delta y_n = y_n - y_{des,0}, \quad (17)$$

$$\Delta \dot{x}_n = \dot{x}_n - \dot{x}_{des,0}, \quad (18)$$

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. 14(a), 14(b), 14(c) and 14(d) show, respectively, the percentage distance $J(n)$ from the desired apex state after $n = 1, 3, 6$ and 9 jumps, on flat terrain, for $\gamma = 20$ and $\sigma_1 = \sigma_2 = 0.8$. Our proposed controller reduces the percentage distance J after 9 jumps from a maximum of about 70% to a maximum of about 0.3%. Fig. 15(a), 15(b), and 15(c) show an example of error reduction for 3 different initial conditions: the error converges to zero.

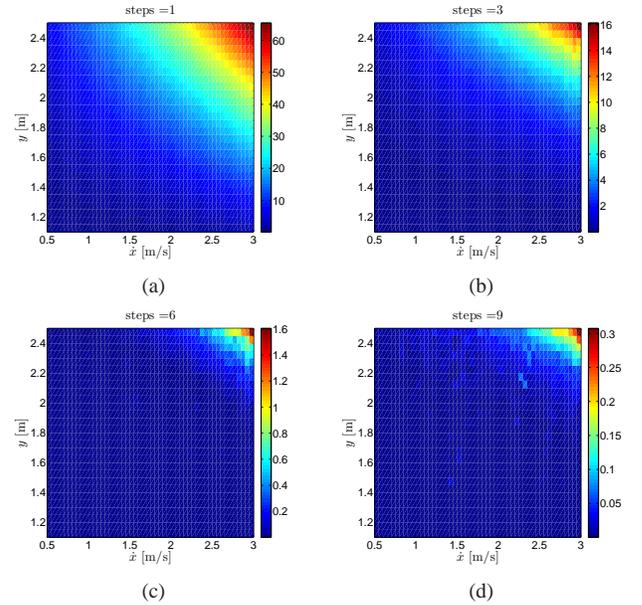


Fig. 14. These plots show the percentage error J after 1 (a), 3 (b), 6 (c) and 9 (d) jumps, for $\gamma = 20$. The x-axis and y-axis represent the apex velocity and the apex height, respectively. Parameters σ_1 and σ_2 have been chosen to be equal to 0.8.

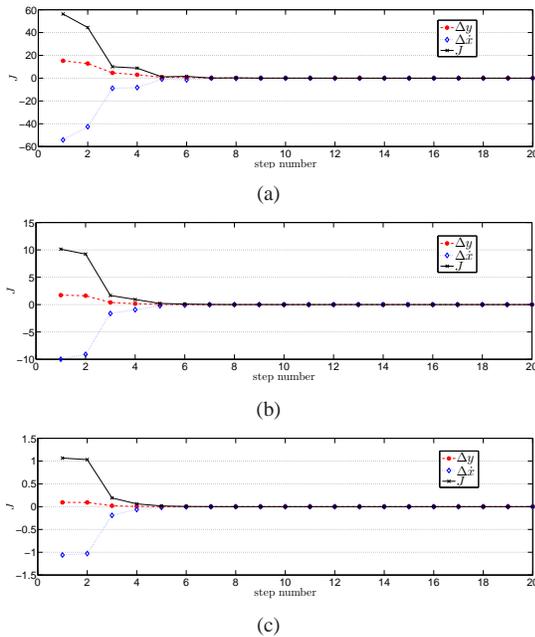


Fig. 15. Evolution at each jump of the apex errors Δy (17) and $\Delta \dot{x}$ (18), and the cost function J (16) for initial and desired apex state $s = \{y_a, \dot{x}_a\} = \{2.4, 2.8\}$ (a), $\{1.8, 1.6\}$ (b), and $\{1.3, 0.8\}$ (c). As we can see, the errors converge to zero.

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, $x_{des,i}^f$, $i = \{1, \dots, N\}$, is allowed, and everything else has to be avoided, as studied in [27]. We can then plan a trajectory to follow based on the knowledge we have of the terrain. In particular, at each jump i we want to find a sequence of control actions θ_{TD} , ℓ_{c2} , and ℓ_{c1} , to minimize the distance between the desired footholds, $x_{des,i}^f$ and the actual landing of the foot, x^f :

$$d_i = \|x_i^f - x_{des,i}^f\|_2,$$

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 jumps we can pre-compute. For a planning horizon of length N , we can write the following cost function to minimize:

$$J_N^f = \sum_{i=1}^N d_i^2.$$

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:

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

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

$$J_N^f = \sum_{i=1}^N d_i^2 + w \sum_{i=1}^N (y_i - y_{des,i})^2 + w_{\dot{x}} (\dot{x}_N - \dot{x}_{N-1})^2, \quad (19)$$

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. 16). However, the horizon length affects the computation 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. 17(a) and 17(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 $N = 2$, with average foothold error over 10,000 jumps 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.

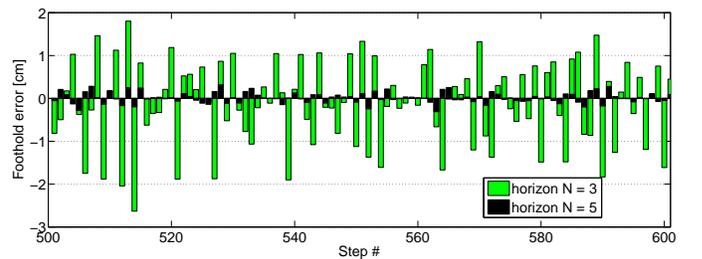


Fig. 16. Example of foothold error for 100 jumps for the case with horizon $N = 3$ (green bars) and $N = 5$ (black bars). As expected, a higher planning horizon gives better performances in terms of error.

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

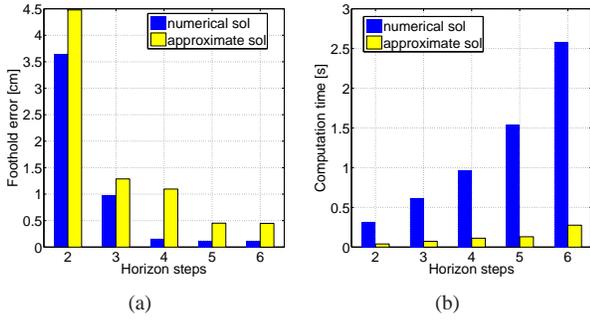


Fig. 17. These plots show the mean foothold error (a) and the mean computation time (b) over 10,000 jumps, for trajectory planning, for different horizon lengths. The terrain has been chosen to be flat, with footholds drawn from a uniform distribution on the open interval $(0.3\ell_0, \ell_0)$. The blue dotted bars refer to the optimization using the numerically computed solution, while the yellow bars refer to the optimization computed using the approximate solution.

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. 9(a) and 9(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 consider 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. 18(a), 18(b), 18(c), and 18(d) show the number of jumps necessary for the system in order to return within 1% of the initial apex state, in the case of positive or negative perturbations. Our controller is robust to perturbations of varying magnitudes, with ability to recover in up to 8 jumps. Clearly, these results are not only due to the control strategy, but are partly affected by actuator limits. One can expect the number of jumps to recover to increase or decrease if the actuator limits are more or less stringent. Fig. 19(a) and 19(b)

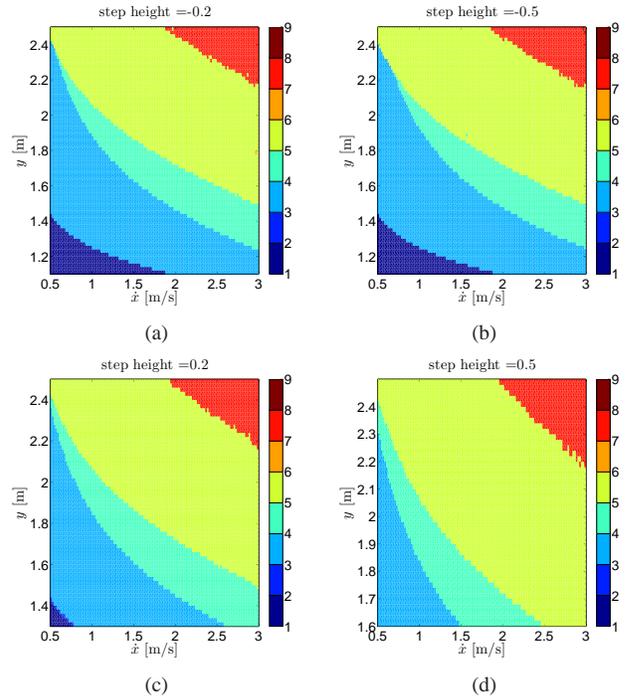


Fig. 18. Number of jumps (colorbar) to reach 1% of desired value, for terrain perturbation of magnitude (a) -0.2 [m], (b) -0.5 [m], (c) 0.2 [m], (d) 0.5 [m]. Simulation parameters are chosen as per Table I. Note that in case of a positive terrain perturbation (subplots (c) and (d)), the apex height y has been chosen to leave enough room to the leg to swing during flight without colliding with the terrain.

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 jump. As we can see from Fig. 20(a), 20(b), and 20(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

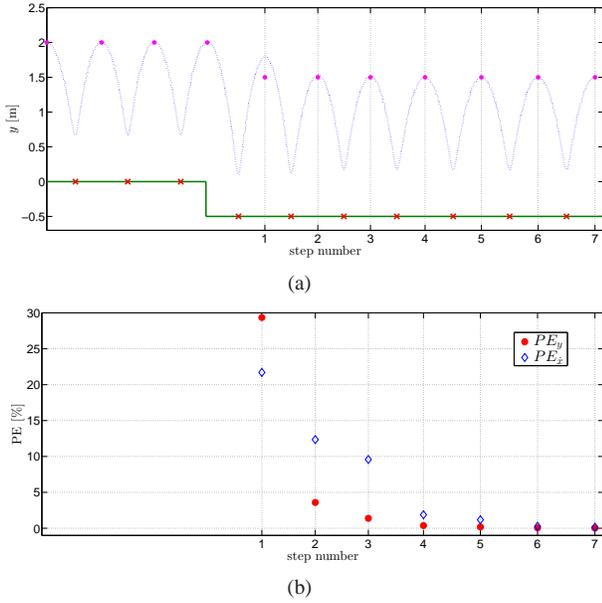


Fig. 19. Plot (a) shows the trajectory of the active SLIP for initial apex state $y = 2$ [m], $\dot{x} = 1.3$ [m/s], and negative terrain perturbation of magnitude 0.6 [m]. The blue dotted line represent the trajectory of the mass, the purple circles the desired apex height. The green solid line is the 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 jump after the drop. We can see that the controller reaches and remains within 1% of the original apex state in 5 jumps.

with what is in Fig. 11(b) and 11(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 jump.

The strategy of recomputing ℓ_{c2} during the first portion of the stance phase after encountering a perturbation can dramatically improve the performance of our controller. Indeed, performance on the same terrain has been studied using the computation during flight only versus with controller update via recomputing of ℓ_{c2} during the first half of stance. Fig. 20(c) shows that the cost function J (computed as in (16)) is significantly smaller than the case without controller update.

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 to be minimized is defined in Eq. (19), 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. 21(a) shows a close-up of the output trajectory, while Fig. 21(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

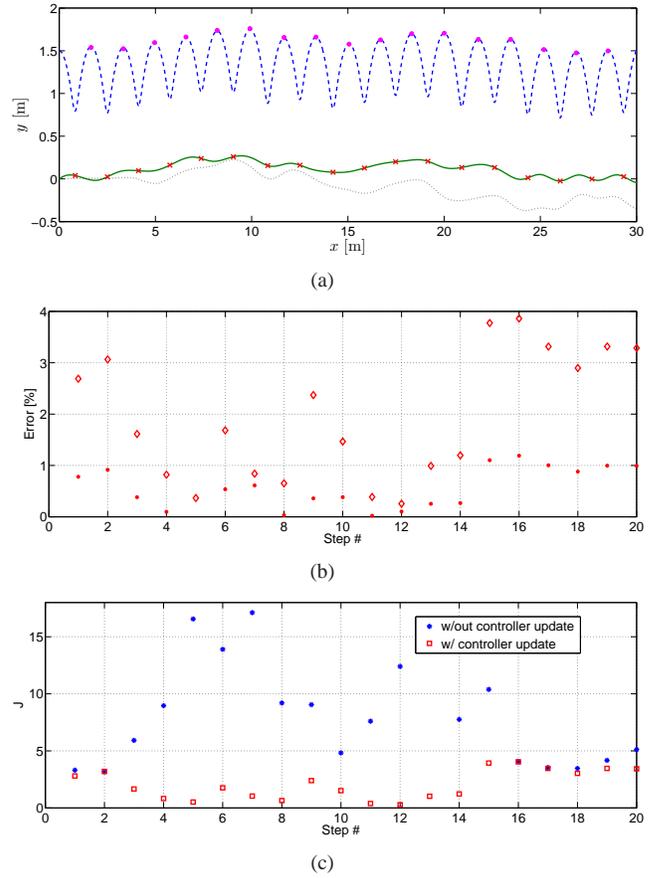


Fig. 20. Plot (a) shows the trajectory of the active SLIP hopping on a random-generated rough terrain. The controller acts to maintain a constant apex height with respect to the terrain of $y = 1.5$ [m], and a constant forward velocity of $\dot{x} = 2$ [m/s]. The blue dotted line represent the trajectory 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 jump. Plot (c) shows the cost J as in (16) for the case without (blue star) and with (red square) controller update during the first part of the stance phase.

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].

D. Extension to general planar hopper models

The actuated SLIP model, while encompassing the general behavior of hopping spring-mass systems, is a highly idealized model. It is then logical to ask how the strategy presented in this work would fare when applied to models that match more accurately real hardware prototypes. To answer this question, the two-step strategy has been tested on a more realistic hopper model, shown in Fig. 22, similar to the hopper studied in [5]. The leg has mass m_l and moment of inertia, J_l , while the body has mass and inertia M and J . Body and leg are connected by a joint at the hip, where an actuator can apply a torque, τ_{hip} . Equations of motion for this system can be easily computed, for example via the Lagrangian method. During flight, a PD controller is applied to the the hip motor to position the leg at the desired angle, and the center of mass of the system follows a ballistic trajectory. The effect of the energy loss at impact on the velocities of the leg length and leg angle is approximated

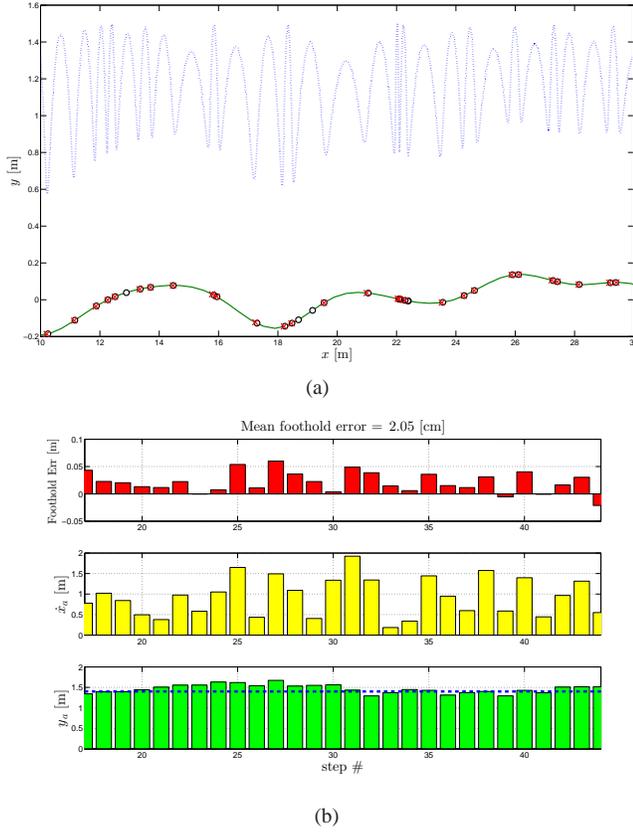


Fig. 21. Plot (a) shows the trajectory of the point mass (blue dotted line) on a terrain (green line). The white circles are the desired footholds, while the red crosses are the actual placements of the foot. As noted in the text, our minimization algorithm allows for some footholds to be skipped. Plot (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.

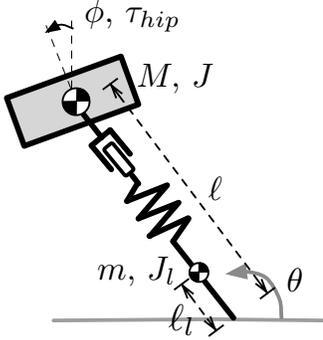


Fig. 22. Planar hopper model. The body has mass M and moment of inertia J , and center of mass at the hip. ϕ is the body angle. The leg has unsprung mass m_l and moment of inertia J_l , and its center of mass is located at distance ℓ_l from the foot. ℓ is the total leg length, and θ the angle that the leg forms with respect to the ground during contact. The spring has stiffness k . ℓ_{act} denotes the length of the series elastic actuator. An actuator at the hip can apply a torque τ_{hip} between the body and the leg.

to determine the initial (post-impact) conditions of the stance phase. During stance, the rotation of the body affects the dynamics of the leg angle, while the dynamics of the leg length is not affected and can be solved as in Section III-A. To keep a body attitude, i.e., to prevent the body from tipping forward or backward during stance, a torque is applied at the

hip. Therefore, an approximation of the equation of motion for the angular displacement over time, $\theta(t)$, is given by the approximation computed in Section III-A, with the addition of a term that approximates the effect of the motion of $\phi(t)$. Note that this is still an approximation, and we exploit the actuator at the hip by applying an additional torque to control the system to follow the approximated trajectory. The series elastic actuator is modelled as a function of input current, with a damping and friction term. Details of the approximation, controllers and parameter values are illustrated in Appendix.

The control strategy has been tested on a flat terrain. Every 10 jumps the target apex height and velocity have been changed to random values taken from a uniform distribution between ± 0.1 [m] and ± 0.3 [m/s] from the previous target state, respectively. Fig. 23(a) and 23(b) show the results for

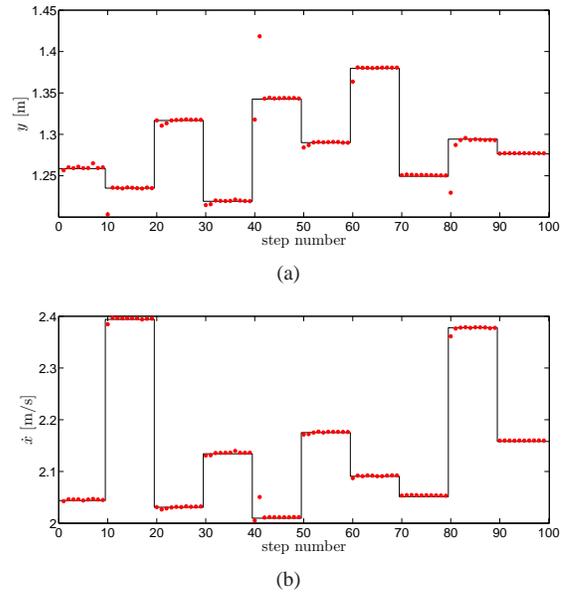


Fig. 23. Data showing the apex state reached for a set of 100 jumps. Plot (a) and (b) show, respectively, the apex height and forward velocity reached (red dots) and the desired values (black line).

100 jumps. The system is controlled to reach simultaneously the apex height and forward velocity desired.

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, and we extend our approximate solution to a more realistic hopper model that includes energy

loss at touch-down, leg inertia, and generally more accurately matches a real hardware prototype.

Starting with the famous Raibert hopper, various studies have been presented on actuated planar hopper models. Our work differs from the previous studies in two main aspects. Firstly, we believe that introducing a more accurate approximation for the stance phase dynamics, and thus freeing us from lengthy numerical computations, allows us to control the system in less jumps that would otherwise be obtained using a classic feedback controller, such as a PID. Additionally, it allows for a prediction and look-ahead of several jumps, which can be usefully applied to foothold placement problems resulting in better accuracy in attaining desired footholds. Secondly, by setting two different actuation targets during the first and second half of the stance phase, it is possible to simultaneously control two out of the three parameters of the following apex state. Our strategy can be applied to more realistic models, and it can be incorporated with existing foothold placement and body attitude control strategies already present in the literature.

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: Neuroethology, 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, Pittsburgh, PA, USA, Oct. 1999.
- [7] U. Saranlı, 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] 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.
- [9] R. M. Ghigliazza, R. Altendorfer, P. Holmes, and D. Koditschek, "A simply stabilized running model," *SIAM Rev.*, vol. 47, no. 3, Mar. 2005.
- [10] S. Riese and A. Seyfarth, "Stance leg control: variation of leg parameters supports stable hopping," *Bioinspiration & Biomimetics*, vol. 7, no. 1, p. 016006, 2012.
- [11] J. K. Hodgins and M. H. Raibert, "Adjusting step length for rough terrain locomotion," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, 1991.
- [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*, Shanghai, China, May 2011, pp. 4000–4006.
- [17] W. J. Schwind and D. E. Koditschek, "Approximating the stance map of a 2 DOF monopod runner," *Journal of Nonlinear Science*, vol. 10, no. 5, pp. 533–588, 2000.
- [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] M. Ahmadi and M. Buehler, "Stable control of a simulated one-legged running robot with a hip and leg compliance," *IEEE Transactions on Robotics*, vol. 13, no. 1, 1997.
- [29] —, "Controlled passive dynamic running experiments with the ARL-Monopod II," *IEEE Transactions on Robotics*, vol. 22, no. 5, 2006.
- [30] 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.

APPENDIX

A. Planar one-legged hopper

The approximation for the stance phase dynamics discussed in Section III-A can be extended and used to predict the dynamics of more realistic planar one-legged hoppers. In particular, we show here how our approximation can be applied to the system shown in Fig. 22 and described in Section VI-D.

The equations of motion for this one-legged hopper can be easily computed from its Lagrangian, with non conservative forces given by the torque at the hip, τ_{hip} . During flight, the center of mass of the system follows a ballistic trajectory, and the servo at the hip moves the body to drive the leg to a desired touch-down angle. The initial conditions for the leg-angle and leg-length dynamics during the stance phase depend on the state of the body at touch-down, $x_{body}(t_{TD})$, $y_{body}(t_{TD})$, and their velocities. These values can be approximated from the

touch-down state of the center of mass as:

$$\begin{aligned} x_{body}(t_{TD}) &\approx x_{com}(t_{TD}) + \frac{m(\ell_0 - \ell_l)}{M + m} \sin \theta_{TD}, \\ y_{body}(t_{TD}) &\approx y_{com}(t_{TD}) + \frac{m(\ell_0 - \ell_l)}{M + m} \cos \theta_{TD}, \\ \dot{x}_{body}(t_{TD}) &\approx \dot{x}_{com}(t_{TD}), \\ \dot{y}_{body}(t_{TD}) &\approx \dot{y}_{com}(t_{TD}). \end{aligned}$$

The energy loss at impact is taken into account at take-off as follows:

$$\begin{aligned} \dot{x}_{com}(t_{TO}) &\approx \frac{M}{M + m} \dot{x}_{body}(t_{TO}) + \frac{m\ell_l}{M + m} \sin(\theta_{TO}) \dot{\theta}_{TO}, \\ \dot{y}_{com}(t_{TO}) &\approx \frac{M}{M + m} \dot{y}_{body}(t_{TO}) + \frac{m\ell_l}{M + m} \cos(\theta_{TO}) \dot{\theta}_{TO}. \end{aligned}$$

During stance, the partial feedback linearization (4) is applied to the system; hence, the closed form solution for the leg length ℓ derived in section III-A is still valid. The series elastic actuator as a function of input current $u_{curr}(t)$ is modelled as:

$$\ddot{\ell}_{act} = \frac{1}{m_{eff}} (-b\dot{\ell}_{act} - \kappa u(t)) - f \text{sign}(\dot{\ell}_{act}), \quad (20)$$

where m_{eff} is the effective mass seen by the actuator, b is the damping constant, f is the friction constant, and κ is the motor constant.

In general, the non-zero momentum of the body affects the dynamics of the leg-angle, $\theta(t)$. Then, θ can be approximated with the solution computed in section III-A, θ_{appr} , with the addition of the term that includes the effect of the body motion over the leg angle, θ_ϕ :

$$\theta(t) = \theta_{appr}(t) + \theta_\phi(t), \quad (21)$$

where

$$\theta_\phi = \frac{J(M + m)}{J_l(M + m) + Mm(\ell - \ell_1)^2} \ddot{\phi}.$$

From the equations of motion, we have that $J\ddot{\phi} = \tau_{hip}$; hence, by choosing an appropriate torque at the hip, θ_ϕ can be analytically solved. In particular, to avoid excessive tipping of the body, τ_{hip} has been chosen as the constant value necessary to keep the body upright. The solution for $\theta(t)$ computed in (21) is an approximation. During stance, a high level PID controller for τ_{hip} is applied to control the system to follow the target dynamics (21).

This approximation has been tested on a path of 100 jumps with varying target apex states, as described in VI-D, with parameters from Table II. Fig. 24(a) show an example of the approximated dynamics of the leg angle during stance compared to their numerical solution, and 24(b) the hip torque required. Fig. 25 shows the series elastic actuator motion, ℓ_{act} and its two components: ℓ_{nl} and ℓ_c , and the current required u_{curr} . In Fig. 26(a) we can see the approximation of the trajectory of the body mass compared to the numerical computed solution, while Fig. 26(b) show the approximation and numerical solution of the forward velocity of the center of mass, for a few jumps.

Simulation Parameters	
$M =$	10 kg
$m =$	1 kg
$J =$	10 kg · m ²
$J_l =$	1 kg · m ²
$\ell_0 =$	1 m
$\ell_l =$	0.5 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
$\gamma =$	20
$u_{curr} \in$	[−20, 20] A
$\tau_{hip} \in$	[−50, 50] N · m
$b =$	50
$f =$	5
$\kappa =$	68.8
$m_{eff} =$	10 kg

TABLE II

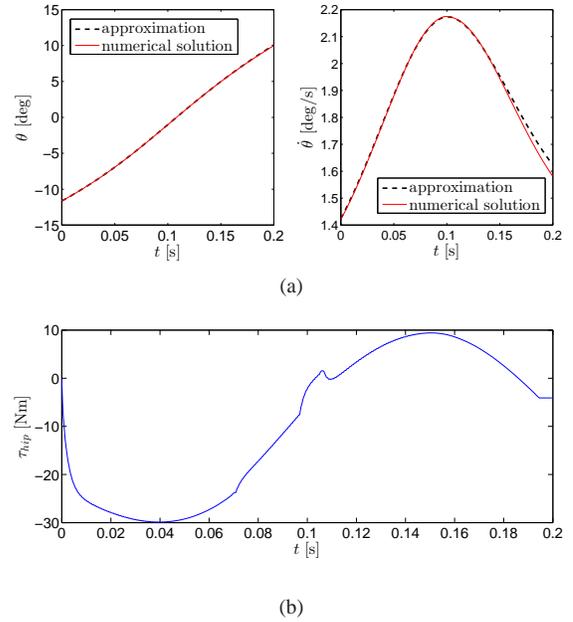


Fig. 24. (a) Data showing the approximated solution of the dynamics of $\theta(t)$ and $\dot{\theta}(t)$ compared to the numerically computed solution. Fig. (b) shows the required torque at the hip.

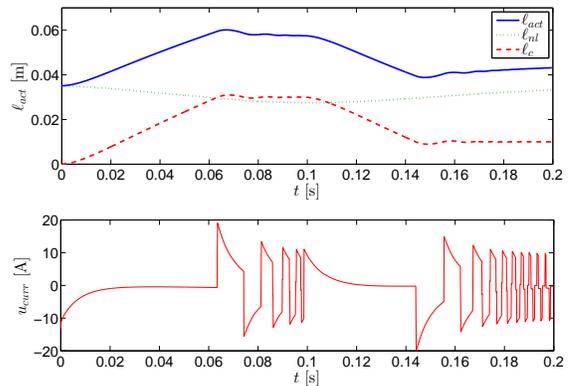
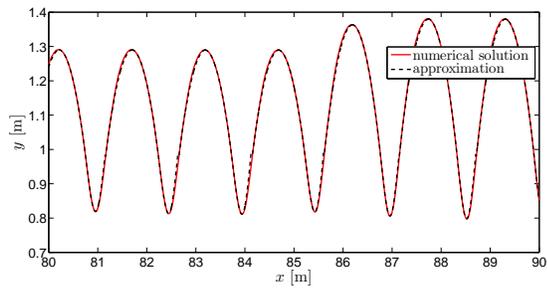
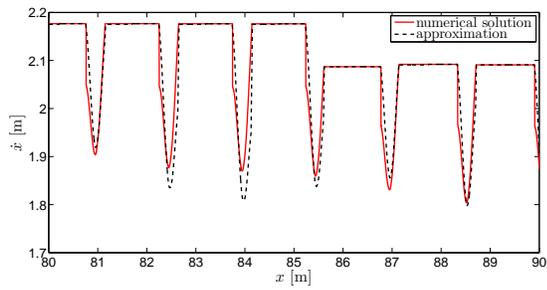


Fig. 25. Series elastic actuator motion $\ell_{act}(t) = \ell_{nl} + \ell_c$ and current u_{curr} . A PID controller drives the actuator dynamics (20) to the target values. The constant values desired during the first and second half of the stance phase are, respectively, $\ell_{c1} = 0.03$, and $\ell_{c2} = 0.01$.



(a)



(b)

Fig. 26. Approximation of the trajectory (a) and forward velocity (b) of the mass compared to the numerically computed solution. The apparent mismatch during the stance phase in \dot{x} is due to the fact that the energy loss at impact is modelled at take-off in the approximated solution.