

Enforced Symmetry of the Stance Phase for the Spring-Loaded Inverted Pendulum

Giulia Piovan and Katie Byl

Abstract—The Spring-Loaded Inverted Pendulum (SLIP) is considered the simplest model to effectively describe bouncing gaits (such as running and hopping) for many legged animals and robots. For this reason, it has often been used as a model for robot design. A key challenge in using this model, however, is the lack of a closed-form solution for the equations of motion that define the stance phase of its dynamics. This results in the impossibility of analytically predicting its trajectory. Consequently, developing a practical control strategy to operate on the model is computationally intensive, because accurately predicting the step-to-step dynamics is still an unsolved problem. By adding an actuator in series with the spring, we can develop a control law for actuator displacement which enforces a desired trajectory during stance. In particular, for our specific chosen control law, we can compute an analytical solution for the stance phase trajectory. Furthermore, we give examples of higher level control strategies for foothold placement and for keeping the forward velocity or the apex height constant on rough terrain that employ our low-level control laws, and we illustrate through simulations the performance typical of our strategy.

I. INTRODUCTION

As is common throughout science, robotics has often looked to nature as a source of inspiration. In the past years, great attention has been focused on legged locomotion for traversal of a wide variety of terrains; however, achieving agility and efficiency comparable to animal locomotion remains an open problem. To better understand the basic dynamics of running and hopping, several studies have centered on the planar Spring-Loaded Inverted pendulum (SLIP). This model keeps the dynamics simple and the number of parameters low. However, the stance phase of the SLIP dynamics is not integrable, resulting in the impossibility of analytically predicting its trajectories, and, consequently, requiring time-costly numerical analysis to model the dynamics accurately.

In this paper, we address the problem of computing a closed-form solution for the SLIP model by enforcing a symmetric gait. The main advantage of having a closed-form solution is to predict exactly the trajectory of the system. Consequently, this simplifies planning for a wide range of control design objectives. A few examples are foothold placement, stability on rough terrain, and the ability to deal with measurement noise and model uncertainty, as discussed further in Sections V and VI.

The SLIP model was first proposed by Blickhan in [1], where the analogy between human hopping and a simple mass-spring system is considered. Subsequently, this model has been widely adopted in biomechanical studies

(e.g. [2] and [3]) and other legged locomotion research, and its dynamics have been embedded in more complex robots. In particular, the study of the trajectory of one-legged bouncing models has been object of several works in the literature, e.g., the famous Raibert hoppers [4], the bow-legged hopping robot [5], and the BiMASC [6]. This increasing interest on the SLIP model has been followed by the development of several control strategies. In [7], the authors propose a control approach to keep the SLIP running speed constant in the presence of uneven ground. No previous knowledge of the terrain is assumed; however, the ability to continuously change the spring stiffness at mid-flight is required, and the strategy does not perform well in the presence of parameter noise. A control strategy that is more robust to model uncertainty and measurement noise has been presented in [8], where the SLIP model is substituted by a simplified hopper approximation, due to the lack of a closed-form solution for the stance trajectory (see [9]). To overcome this limitation, several researchers have tried to obtain an analytical approximation to the nonlinear model dynamics (see [10], [11], and [12]). In particular, in [11] an approximation is given, and in [12] a gravity correction scheme is added to it in order to increase its accuracy for non-symmetric SLIP trajectories. From Fig. 4 in [12] we can see that the mean error percentage for the prediction of the apex position and of the take-off velocity for a leg length of $1[m]$, respectively, about 7% and 12% for [11], 6% and 8% for [10], and 3% and 5% for [12]. Since the error propagates at each step, the development of a control law for motion planning given a limited set of feasible footholds is not generally practical. In [13], the authors incorporate a leg actuation in series with the spring to the classic SLIP model to add variations in the system energy during stance, in order to produce asymptotically stable gaits.

This paper contains several contributions. First, we modify the classic SLIP model by adding an actuator, in series with the spring, that has the ability to compress/decompress the spring during stance. Second, we develop a control law for actuator displacement, to enforce a desired trajectory during stance. Third, we chose a particular desired symmetric trajectory for the stance phase, for which we compute an analytical solution. Finally, we extend the solution to non-symmetric trajectories by allowing the leg to leave the ground either before or after the state which mirrors the touch-down state has been achieved.

The rest of the paper is organized as follows. Section II reviews the structure and dynamics of the SLIP model and it contains symbols and definitions that will be used throughout the paper. Section III presents the problem setup and the computation of the analytical solution for the desired

G. Piovan and K. Byl are with the Center for Control, Dynamical Systems and Computation, University of California at Santa Barbara, Santa Barbara, CA 93106, USA, giulia@engineering.ucsb.edu, katiebyl@ece.ucsb.edu

symmetric trajectory. Section IV explores the two main control actions available, and exploits the possibility of adding/removing energy to the system. Section V contains some useful applications, and Section VI shows the validity of the discussed applications through simulations. Finally, Section VII contains conclusions and future work.

II. PRELIMINARIES

A. Passive SLIP model

The SLIP is modelled as a point mass attached to a massless spring leg. The terminal part of the leg is referred to as the *foot*. The dynamics are divided into two phases. The *flight phase*, where the body follows a ballistic trajectory; and the *stance phase* (1)-(2), where the mass dynamics are affected by the compression and decompression of the spring:

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

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

where g is the gravitational acceleration constant, m is the body mass, k is the spring stiffness constant, ℓ_k is the spring length, and $\ell_{k,0}$ is the spring length at equilibrium. Furthermore, $\ell(t)$ is the leg length as a function of time, and $\alpha(t)$ is the leg angle measured clockwise with respect to the negative horizontal axis, as shown in Fig. 1(a). At

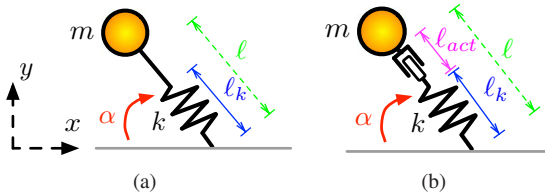


Fig. 1. Plot 1(a) illustrates the classic SLIP model (passive SLIP), and plot 1(b) shows the active SLIP model (active SLIP)

each instant in time, we can define the state of the mass in Cartesian coordinates $s(t) = \{x(t), y(t), \dot{x}(t), \dot{y}(t)\}$, where

$$x(t) = -\ell(t) \cos \alpha(t), \quad y(t) = \ell(t) \sin \alpha(t),$$

and \dot{x} and \dot{y} are their time derivatives.

Let us now introduce some definitions and nomenclature that will be used throughout the paper. The ballistic trajectory can be divided in two parts: *descent* and *ascent*, where the mass vertical velocity is negative or positive, respectively (see Fig. 2). The transition from flight to stance and vice-versa is characterized by two events. The instant when the foot first touches the ground is called *touch-down (TD)*, which marks the transition between descent and stance. The instant when the foot leaves the ground is called *take-off (TO)*, which marks the transition between stance and ascent. The highest point (in terms of vertical position y) reached by the mass during the ballistic trajectory is called the *apex state*, and it divides the ascent and descent parts of flight. By definition, the vertical velocity at apex is $\dot{y}_a = 0$ [m/s]. Hence, the apex state is completely defined by a three-dimensional vector $s_a = \{x_a, y_a, \dot{x}_a\}$. We call a *jump* the transition between one apex state, $s_a(\kappa)$, and

the next, $s_a(\kappa + 1)$. We say that a jump is *successful* if the forward velocity during the entire trajectory only takes positive values, and the angle at take-off is $\alpha_{TO} \in [\pi/2, \pi]$. Furthermore, to allow the leg to swing forward in flight, we require the distance between $y_a(\kappa + 1)$ and the terrain height at $x_a(\kappa + 1)$ to be bigger than the leg length at equilibrium.

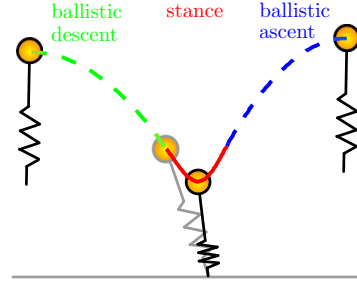


Fig. 2. Scheme that illustrates the phases of the SLIP model trajectory

Despite its simple structure, the SLIP stance dynamics are described by non-integrable equations (1)-(2). Therefore, it is in general not possible to predict the trajectory of the mass, and consequently it is not an easy task to implement control strategies for successful hopping in the presence of noise and/or rough terrain.

Note that, for the rest of the paper, we will refer to the SLIP model introduced above as the *passive SLIP*.

III. PROBLEM SETUP

A. Active SLIP model

Now, consider the passive SLIP, and let us add to the leg an actuator in series with the spring, as shown in Fig. 1(b). Such actuator is piston-like, and we assume it can extend or retract within a certain displacement range throughout the trajectory. The purpose of the actuator is to perform an additional compression/decompression to the spring, in order to modify the natural dynamics of the system. Several studies, e.g. [14] and [15], provide evidence that legs in animal can store and return energy, and that the ability of legs to produce, storage and dissipate energy is critical in locomotion.

Let $\ell_{act}(t)$ be the actuator length, and $\ell_{act,0}$ be the actuator length at rest. The total leg length is then given by $\ell(t) = \ell_{act}(t) + \ell_k(t)$. Define the actuator displacement as $\Delta \ell_{act} = \ell_{act} - \ell_{act,0}$. Then, from equation (1), the actuator displacement necessary to maintain a desired feasible trajectory $\alpha(t)$ and $\ell(t)$ during the stance phase, is given by:

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

where $\ell_0 = \ell_{k,0} + \ell_{act,0}$.

Note that we will refer to the SLIP model with actuator as the *active SLIP*.

B. Enforced symmetric gait

The control action required to enforce a symmetric gait to the SLIP model is not unique. However, not all possible symmetric laws can be solved in closed form. In what follows, we start by observing that the passive natural dynamics

of the angular velocity with respect to the angular position, has a shape that is similar to a part of a sine wave (see Fig. 3 (c)). Therefore, a logical approach is to enforce the following dynamics to the angular velocity:

$$\dot{\alpha}(t) = A \sin \alpha(t) + c_1, \quad (4)$$

where the parameters A and c_1 are uniquely determined such that position and velocity of the mass at the start of the stance phase match the touch-down state. Substituting (2) in the time derivative of (4) gives:

$$A = -\frac{2\dot{\ell}_{TD}\dot{\alpha}_{TD} + g \cos \alpha_{TD}}{\ell_{TD}\dot{\alpha}_{TD} \cos \alpha_{TD}},$$

$$c_1 = \dot{\alpha}_{TD} - A \sin \alpha_{TD},$$

where α_{TD} , $\dot{\alpha}_{TD}$, ℓ_{TD} , $\dot{\ell}_{TD}$ are position and velocity at touch-down of leg angle and length, respectively. We can then solve (1) to find a closed-form solution for the leg length dynamics:

$$\ell(t) = \frac{g}{A\dot{\alpha}(t)} + \frac{c_2}{\sqrt{\dot{\alpha}(t)}}, \quad (5)$$

with

$$c_2 = \ell_{TD}\sqrt{\dot{\alpha}_{TD}} - \frac{g}{A\sqrt{\dot{\alpha}_{TD}}}.$$

It is important to note that the leg length $\ell(t)$ can never take negative values. Since the minimum value of $\ell(t)$ corresponds to the case $\alpha(t) = \pi/2$, then

$$\frac{g}{A(A+c_1)} + \frac{c_2}{\sqrt{A+c_1}} > 0.$$

As for (5), we can solve (4) to find a closed-form solution for the angle dynamics during stance:

$$\alpha(t) = -2 \tan^{-1} \left(\frac{A + c_4 \tanh(c_4(t + c_3)/2)}{c_1} \right), \quad (6)$$

for $t \in [0, t_{TO}]$, with

$$c_3 = -\frac{2}{c_4} \tanh^{-1} \left(\frac{A + c_1 \tan(\alpha_{TD}/2)}{c_4} \right),$$

and

$$c_4 = \sqrt{A^2 - c_1^2}. \quad (7)$$

The take-off time, t_{TO} , is given by:

$$t_{TO} = -c_3 - \frac{2}{c_4} \tanh^{-1} \left(\frac{A + c_1 \tan(\alpha_{TO}/2)}{c_4} \right),$$

where α_{TO} is the take-off angle. In order for the above equations to be feasible, we require the following constraint:

$$|A + c_1 \tan(\alpha_{TD}/2)| \leq |\sqrt{A^2 - c_1^2}|,$$

and the same constraint can be enforced for all $\alpha(t)$, obtaining the following:

$$\alpha(t) \in [2 \tan^{-1} \left(\frac{-c_4 - A}{c_1} \right), 2 \tan^{-1} \left(\frac{c_4 - A}{c_1} \right)]. \quad (8)$$

Note that the parameter c_4 , defined in (7), can be a real or imaginary number. All equations from (6) to (8) still hold.

Plots in Fig. 3 show an example of the mass trajectory and the angular acceleration of passive and active SLIP for the same initial conditions, and the required actuator displacement for the active case.

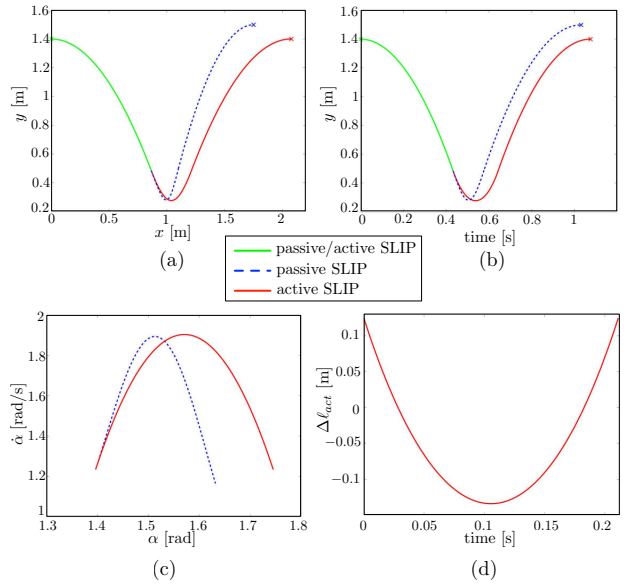


Fig. 3. In these plots, the blue dotted line refers to the passive SLIP, while the solid red line refers to the active SLIP. Figs. (a) and (b) show the mass trajectory for passive and active SLIP with respect to the horizontal position and with respect to time, respectively. Fig. (c) shows the evolution of $\dot{\alpha}$ during stance for the passive and active SLIP, and Fig. (d) represents the required actuator displacement $\Delta \ell_{act}$. The parameters used are $k/m = 500[s^{-2}]$, $\ell_0 = 1[m]$ and $\ell_{k,0} = 0.5[m]$. The initial apex state is $\mathbf{s}_a = \{0[m], 1.4[m], 2[m/s]\}$, the touch-down angle is $\alpha_{TD} = 80[deg]$, and terrain height is $0[m]$.

IV. CONTROL ACTIONS

A. Fixed leg length at take-off

When the trajectory is symmetric, we have that $\alpha_{TO} = \pi - \alpha_{TD}$, and the apex velocity components, \dot{x}_a and \dot{y}_a , and the mass apex height y_a are preserved from one apex state to the next. The only control action available is the touch-down angle α_{TD} , and its choice affects the forward position x_a at the next apex state. If we start from the same initial apex state and we use the same set of possible touch-down angle α_{TD} , it is clear that enforcing a symmetric trajectory changes the set of reachable apex states with respect to the set reachable by the passive SLIP. As an example, Fig. 4 shows the set of reachable apex states for the active and passive SLIP. We can see that, despite the different spatial direction of the two curves, the span of reachable points is comparable in size. This suggests that, even if the reachable apex states are not the same, enforcing a symmetric gait does not generally represent a limitation with respect to the passive SLIP, but it rather changes the direction of the reachable space.

B. Variable leg length at take-off

There are some cases when enforcing a symmetric gait can pose a limit on the capability of successfully undergoing a particular terrain. For example, on terrains with tall obstacles, or with stairs, it may be necessary to add (or remove) energy to the system to achieve a successful jump. Therefore, we allow the leg to leave the ground either before or after the state which mirrors the touch-down state has been achieved. This simply requires setting $\Delta \ell_{act}$ such that the leg spring is fully unloaded, i.e., such that $\Delta \ell_{act} = \ell - \ell_0$. It is then possible to chose the leg length at take-off, or equivalently to

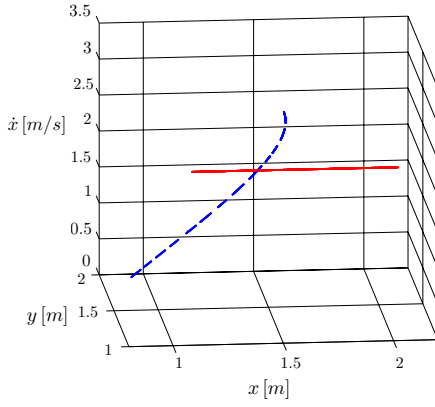


Fig. 4. The dotted blue line and the solid red line represent the reachable points for passive and active SLIP, respectively, with touch-down angle $\alpha_{TD} \in [10, 90]$ [deg]. The system parameters are $k/m = 500[s^{-2}]$, $\ell_0 = 1[m]$, initial apex state $\mathbf{s}_a = \{0[m], 1.4[m], 2[m/s]\}$, and terrain height $0[m]$. Note that both lines lie on a surface of constant energy.

chose the take-off angle α_{TO} to be different than $\pi - \alpha_{TD}$. Hence, the mass will leave the ground at a different take-off state, $(x_{TO}, y_{TO}, \dot{x}_{TO}, \dot{y}_{TO})$, and will reach a different new apex state than otherwise reached. In particular, the velocity components at take-off, \dot{x}_{TO} and \dot{y}_{TO} , will be different than the velocity components at touch-down, \dot{x}_{TD} and \dot{y}_{TD} . Fig. 5 shows the set of all the apex states reachable from an initial nominal apex state. The coloring scheme reflects the direction of the next apex in state space with respect to controlling the touch-down angle α_{TD} , Fig. 5(a), or the leg length at take-off ℓ_{TO} , Fig. 5(b).

The main advantages of such a procedure are two. First, the capability of adding/subtracting energy to the system allows us to reach apex states otherwise unreachable, which converts into the ability of operating in a wider variety of terrains. Second, while choosing α_{TD} is a control action that happens before the impact with the terrain, the choice of α_{TO} is done *after* the impact. This represents a big advantage in case of noisy measurements (e.g., on the terrain, on the position of the leg, etc.), since it is now possible to partially act against the effect of the noise during the actuated stance phase.

V. APPLICATIONS

This section contains two applications of the proposed symmetric enforcement strategy. Note that the applications are manifold, and here below we just show two useful examples. In the first application, we assume the terrain is divided into allowed and not allowed foothold sets, and we want to determine a path from an initial foothold to an end foothold. In the second application, we aim to reach a desired constant forward velocity or constant apex height during motion on unknown rough terrain. We will assume that the actuator can be extended/contracted only up to a certain percentage of its equilibrium length, and define with ℓ_{act}^{max} the maximum feasible actuator length. This will pose a constraint on the velocity/height reachable in one step.

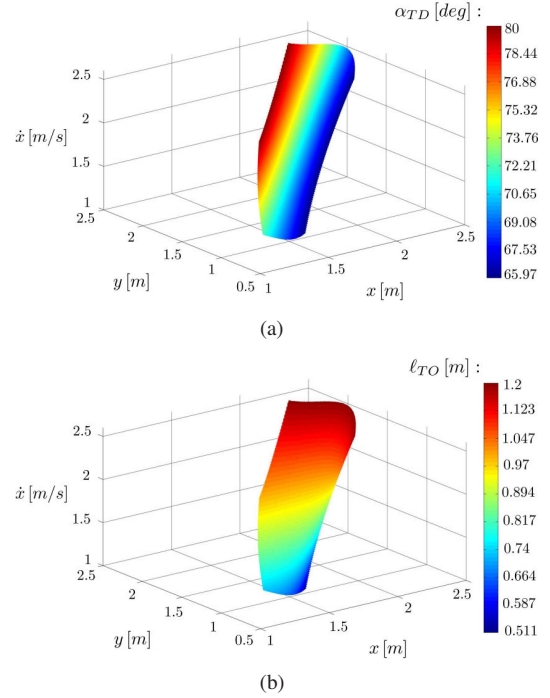


Fig. 5. These plots represent the set of apex states reachable from the initial apex state $\mathbf{s}_a = \{0[m], 1.4[m], 2[m/s]\}$ in the actuated case with leg length at take-off $L_{TO} \in [0.5, 1.2]$ [m] and $\alpha_{TD} \in [63, 80]$ [deg]. The system parameters are $k/m = 500[s^{-2}]$, $\ell_0 = 1[m]$, and the terrain height is $0[m]$.

A. Foothold placement

One of the main applications of the closed-form solution, is the ability to plan a trajectory given a set of feasible footholds: for example, trajectory planning on a terrain with forbidden areas, such as holes or water.

For any initial apex state $\{x_a, y_a, \dot{x}_a\}$ and terrain height h , we can identify a minimum and a maximum touch-down angle, α_{min} and α_{max} , that allow the system to complete a successful jump. Let us call *step length* the distance between two subsequent landing points, and let us define the two quantities m and M as

$$\begin{aligned} m &= \dot{x}_a \sqrt{\frac{2}{g}(y_a - h - \ell_0 \sin \alpha_{max})} + \ell_0 \cos \alpha_{max}, \\ M &= \dot{x}_a \sqrt{\frac{2}{g}(y_a - h - \ell_0 \sin \alpha_{min})} + \ell_0 \cos \alpha_{min}. \end{aligned} \quad (9)$$

When the gait is symmetric, the minimum and maximum step lengths at each apex state $\{x_a, y_a, \dot{x}_a\}$ depend on the length of the previous step as follows: the minimum step length is $s_m = d_0 + m$, and the maximum is $s_M = d_0 + M$, where d_0 is the distance between x_a and the previous landing point. It follows that the distance between two feasible footholds has to take values between $[d_0 + m, d_0 + M]$. It is straightforward to show that d_0 is bounded: $d_0 \in [m, M]$. Therefore, a necessary but not sufficient condition for foothold placement is that the distance between two feasible footholds has to be between $[2m, 2M]$. It is clear that the required distance between footholds is not an absolute measure, but it depends on the previous step length and, in the case of rough terrain,

on the terrain height. In fact, values (9) depend on the terrain height at the landing point, h . For simplicity, we will assume the terrain to be flat; however, what follows can be easily adjusted in case of rough terrain.

Assume now that the terrain is divided into allowed and not allowed foothold sets, and that we want to determine a path from one foothold to another one v_{end} (or to a set of footholds, \mathcal{V}_{end}), given an initial apex state and initial d_0 . We also assume that, at each step, we have the capability to see the terrain up to a certain distance D from our current location. Let us create a graph where each vertex represents a foothold. Starting from an initial foothold/vertex v_0 , and initial length d_0 , if $D \geq d_0 + m$ we add an edge from v_0 to all the footholds v_i whose forward distance from v_0 is $d(v_0, v_i) \in [d_0 + m, \min\{D, d_0 + M\}]$, and we associate to the edge connecting v_0 and v_i the length $d_i = d(v_0, v_i) - d_0$. (Note that, if $D < d_0 + m$, there is no feasible foothold for that specific initial condition.) Then, for each vertex v_i , we repeat the same process, until all footholds have been explored. As output, we obtain a graph for each feasible path from node v_0 to node $v \in \mathcal{V}_{end}$. The optimal path can then be chosen based on the specific definition of cost function: e.g., we can associate a cost to each edge based on the length covered, or based on the probability that the foothold location is correctly sensed to be feasible.

Algorithm 1 compute feasible paths

- 1: $\mathcal{V} = \{f_0, \dots, f_n\}$ set of all footholds
 - 2: $\mathcal{V}_{end} \subseteq \mathcal{V}$ set of desired path end
 - 3: Initial state $v = f_0$, initial length d_0
 - 4: graph $G = (V, E, W)$, with $V = \{v\}$, $E = \emptyset$, $W = \emptyset$.
 - 5: **for** $v_i \in \mathcal{V}$ s.t. $d(v_i, v) \in \min\{D, [d_0 + m, d_0 + M]\}$
do
 - 6: create edge e_{v,v_i} with weight $d_i = d(v, v_i) - d_0$
 - 7: cost C_i
 - 8: $V_i := \{V, v_i\}$, $E_i := \{E, e_{v,v_i}\}$, $W_i := \{W, d_0\}$
 - 9: $G_i := \{V_i, E_i, W_i\}$
 - 10: Repeat steps 5–10 for $v = v_i$, $d_0 = d_i$, and $G = G_i$
 until no more possible forward jump is feasible
 - 11: **end for**
 - 12: **return** all graphs G that end in \mathcal{V}_{end}
-

The complexity of Algorithm 1 grows with the amount of knowledge of the terrain available and the number of feasible footholds. However, we can intuitively suspect that, when the feasible regions of the terrain are high in number and equally spaced, a complete knowledge of the terrain is not required. Such algorithm is practical if jump transitions can be computed quickly, therefore having an analytic solution is a necessary condition to keep the computation complexity low. We will not discuss further the relationship between the complexity of our algorithm and the terrain knowledge D , being beyond the purpose of this paper.

B. Constant forward velocity/constant apex height

Here, we consider two applications for our variable take-off leg length strategy, concerning the problems of reaching a desired running speed and of reaching a desired apex height with respect to the terrain. In particular, the problem of

reaching a desired running speed has already been studied in the literature with different assumptions on the model (see [7] and [8]).

First, let us consider the problem of reaching a desired running speed at the next apex state, \dot{x}_d , on rough terrain. Given the initial touch-down conditions (α_{TD} and ℓ_{TD}), we want to compute the take off angle that allow us to obtain the desired forward velocity, i.e., we need to solve the following equation:

$$\dot{x}_d = -\dot{\ell}_{TO} \cos \alpha_{TO} + \dot{\alpha}_{TO} \ell_{TO} \sin \alpha_{TO}. \quad (10)$$

Define the variable $z = \sqrt{A \sin \alpha_{TO} + c_1}$. After some tedious calculations, one can show that solving equation (10) is equivalent to solving for z the following equation:

$$Ac_2 z^5 + (2c_1 g - 2A^2 \dot{x}_d) + z(A^3 c_2 - Ac_1^2 c_2) - 2c_1^2 g + 2A^2 g = 0,$$

which cannot be solved analytically but can be easily solved numerically. In order to obtain \dot{x}_d , the leg needs to leave the ground at an angle $\alpha_{TO} = \pi - \sin^{-1}(\frac{z^2 - c_1}{A})$, corresponding to a certain take-off leg length.

Since the actuator ability to contrac/extend is limited to ℓ_{act}^{max} , then $\ell_{max} = \ell_{act}^{max} + \ell_k$. From equation (5) we obtain

$$\frac{g}{A \dot{\alpha}_{TO}} + \frac{c_2}{\sqrt{\dot{\alpha}_{TO}}} \leq \ell_{max},$$

and therefore

$$z \geq \frac{c_2 + \sqrt{c_2^2 + 4g\ell_{max}/A}}{2\ell_{max}}. \quad (11)$$

By defining

$$\hat{z} = \frac{c_2 + \sqrt{c_2^2 + 4g\ell_{max}/A}}{2\ell_{max}}, \quad c_5 = \frac{\hat{z}^2 - c_1}{A},$$

we can compute the maximum forward velocity obtainable in one step as:

$$\begin{aligned} \dot{x}_{max} &= \dots \\ &= \left(\frac{g}{A \hat{z}^2} + \frac{c_2}{\hat{z}}\right) \left(\frac{A}{2} - \frac{Ac_2^2}{2} + \hat{z}^2 c_5\right) + \frac{g}{2 \hat{z}^2} (1 - c_5^2). \end{aligned} \quad (12)$$

The minimum forward velocity can be easily computed as the value the velocity takes at the time \hat{t} when $\alpha(\hat{t}) = \pi/2$.

Now, let us consider the problem of maintaining a constant apex height with respect to the ground. This application is particularly useful in situations where the terrain height is progressively increasing and/or decreasing. In fact, it is crucial to maintain a certain minimum distance from the terrain to avoid collision of the leg with the ground. At the same time, it is important to keep the relative apex height limited to avoid gaining excessive velocity during the ballistic descent, to avoid ‘bottoming out’ the passive spring, with the resulting impossibility to perform a successful jump.

Let us assume no prior knowledge of the terrain, in which case our strategy focuses on keeping the height of the next apex constant with respect to the ground at the previous landing point, which is known a posteriori. Given the initial touch-down conditions (α_{TD} and ℓ_{TD}) and the terrain height at the previous foothold, y_t , we want to compute the take-off angle that results in obtaining the desired apex height Δy_d with respect to y_t . Note that $\Delta y_d = y_a = y_t$, where y_a is

the next apex height. Solving this problem is equivalent to solving for α_{TO} the following equation:

$$\Delta y_d = \ell_{TO} \sin \alpha_{TO} + \frac{1}{2g} \left(\dot{\ell}_{TO} \sin \alpha_{TO} + \ell_{TO} \dot{\alpha}_{TO} \cos \alpha_{TO} \right)^2. \quad (13)$$

As for the constant velocity case, this is equivalent to solve for $z = \sqrt{A \sin \alpha_{TO} + c_1}$ a 10th-order polynomial equation in z (not shown here because of its excessive length). Because of limitation on the actuator length, we obtain again constraint (11) on z . Correspondingly, we can compute the maximum and minimum apex height obtainable in one step.

VI. SIMULATIONS

In what follows we illustrate through simulations the performance of our example control strategies as discussed in V-A and V-B. The system parameters used are defined in the following table:

$k/m [s^{-2}]$	$g [m/s^2]$	$\ell_{act,0} [m]$	$\ell_{k,0} [m]$
100	9.81	0.5	0.5

A. Foothold placement

The following simulations refer to V-A. Algorithm 1 has been tested on a rough terrain of length 50 [m]. Knowledge of the terrain ahead has been assumed to be $D = 10$ [m]. The terrain height has been built as a normal distribution with 0 mean and standard deviation $\sigma = .1$. The terrain step length has been chosen to be 0.6 [m]. The apex state is $s_a = \{1.3, 1.4, 3\}$ and $d_0 = 1.3$ [m]. In order to use the algorithm, we discretized the terrain in intervals of size 0.1 [m], and we considered to be unfeasible all the points x_t with height $y_t(x_t)$ such that $|y_t(x_t)| \geq 0.2$, and all the parts close to a change of height, in order to avoid leg collision with the terrain during stance. The cost function has been chosen to be the final covered distance. The output is shown in Fig. 6. The green and red line represent respectively the feasible and unfeasible parts of the terrain, the blue dotted line represents the mass trajectory, and the black crosses are the landing points.

B. Constant forward velocity/constant apex height

We now demonstrate through simulations performance typical of the control strategies introduced in V-B.

First, let us examine the problem of reaching a desired forward speed. Consider the active SLIP model, and assume the actuator on the leg can extend/contract up to 20% of the leg equilibrium length. Let us assume the active SLIP travels on a rough terrain, but that it is not provided with any knowledge of it. We will then treat the terrain as if it were completely flat. We also fix the touch-down angle at $\alpha_{TD} = 80$ [deg]. During stance, we compute the take-off angle to reach the desired velocity by solving (10) as follows. If the maximum forward velocity obtainable in one step, \dot{x}_{max} computed with (12), is $\dot{x}_{max} \geq \dot{x}_d$, then the take-off angle α_{TO} will be computed using (10). If $\dot{x}_{max} < \dot{x}_d$, then α_{TO} will be chosen to reach at the next apex forward velocity \dot{x}_{max} , and therefore the number of steps necessary to reach the desired steady-state will be greater than one. Fig. 7(a) shows a numerical example, with initial apex state $s_a = \{0[m], 1.3[m], 2[m/s]\}$ and desired forward velocity $\dot{x}_d =$

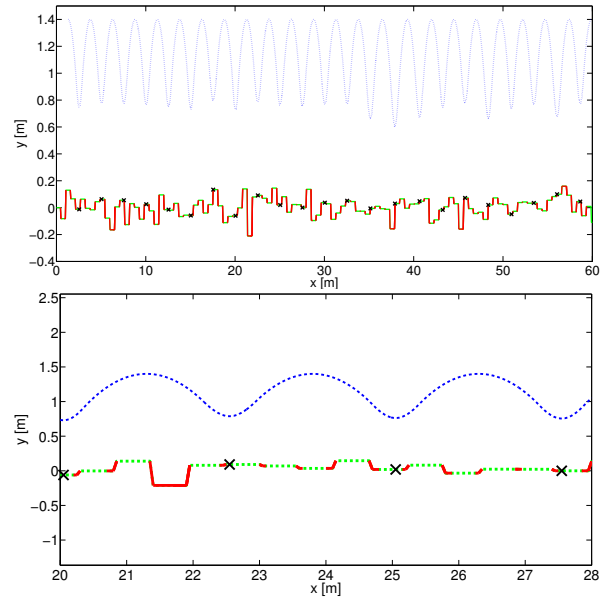


Fig. 6. The figure shows the path computed by algorithm 1 where the cost function is defined to be the final covered distance. The green and red line represent respectively the feasible and unfeasible parts of the terrain, the blue line represents the mass trajectory, and the black crosses are the landing points.

2.5[m/s]. As we can see, the system reaches the desired forward velocity in 2 steps, and it is able to successfully run on a rough terrain with error that often exceeds 50% of the leg length. Fig. 7(b) shows a numerical example, with initial apex state $s_a = \{0[m], 2.5[m], 2.3[m/s]\}$ and desired forward velocity $\dot{x}_d = 2[m/s]$, smaller than the initial apex velocity. As we can see, the system reaches the desired forward velocity in 1 step.

Let us now examine the problem of reaching a desired height from the terrain, assuming the actuator can extend/contract up to 20% of the leg equilibrium length. Let us assume the active SLIP is not provided with any knowledge of the terrain it travels on. We fix the touch-down angle at $\alpha_{TD} = 80$ [deg]. During stance, we compute the take-off angle to reach the desired terrain height with respect to the previous landing height by solving (13). Fig. 8(a) shows a numerical example, with initial apex state $s_a = \{0[m], 2.3[m], 2.3[m/s]\}$ and desired relative height $\Delta y_d = 1.5[m]$, whereas Fig. 7(b) shows a numerical example, with initial apex state $s_a = \{0[m], 1.3[m], 2[m/s]\}$ and desired relative height $\Delta y_d = 1.7[m]$.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we modified the classic SLIP model by adding an actuator in series with the spring. We developed a control law for actuator displacement in order to enforce a symmetric stance phase to the model. For the specific control law we enforced, we were able to compute a closed-form solution for the trajectory of the mass during the stance phase. We then exploited the possibility of the leg leaving the ground either before or after reaching the take-off state that is symmetric to the touch-down state, and as a consequence we were able to add and remove energy from the system. We also provided two examples of useful applications for

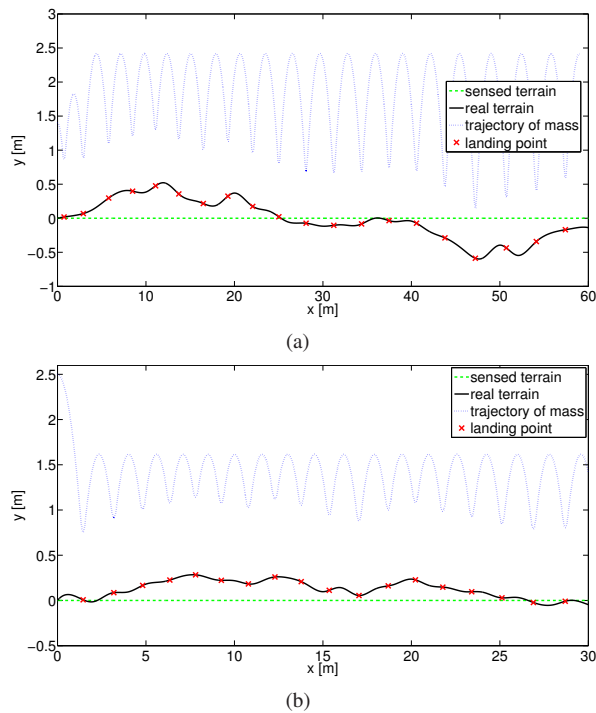


Fig. 7. The figure shows two numerical examples of the forward velocity control strategy proposed in V-B. The black line represents the real terrain, and the dotted green flat line the sensed terrain. The red crosses are the landing points and the dotted blue line is the mass trajectory.

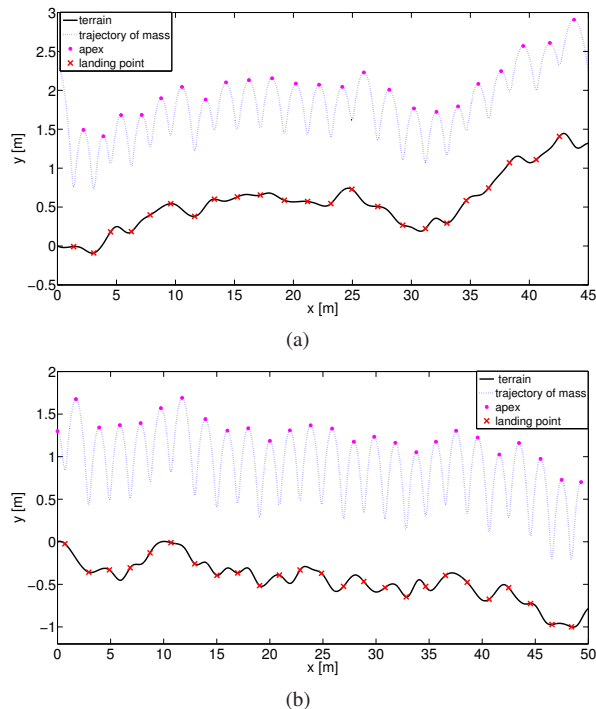


Fig. 8. The figure shows two numerical examples of the relative height control strategy proposed in V-B. The black line represents the real terrain, the dotted blue line is the mass trajectory, the red crosses are the landing points and the magenta dots are the apex states.

our method (trajectory planning on fixed footholds, and running on rough terrain at constant forward velocity and

at a constant relative height), and we validated them through simulations.

We are currently extending the work in several directions. First, we want to expand our control strategies by developing a control law for optimally positioning the leg at touchdown in the presence of rough terrain. Second, we aim to extend the analysis of our current strategy for foothold placement. In particular, we want to characterize, for several terrain types, what amount of knowledge of the terrain it is required to successfully compute a feasible path, and how this affects the computational effort. Third, we wish to adapt our modeling approach to more accurately match a real hardware prototype; and to verify our theoretical ideas on a hopping robot developed in our lab.

ACKNOWLEDGMENTS

This work is supported in part by DARPA (Contract No. W911NF-11-1-0077) and by the Institute for Collaborative Biotechnologies (Grant No. W911NF-09-0001) from the U.S. Army Research Office.

REFERENCES

- [1] R. Blickhan, "The spring-mass model for running and hopping," *Journal of Biomechanics*, vol. 22, no. 11/12, pp. 1217–1227, 1989.
- [2] R. Blickhan and R. J. Full, "Similarity in multilegged locomotion: Bouncing like a monopode," *Journal of Comparative Physiology A: Neurothology, Sensory, Neural, and Behavioral Physiology*, vol. 173, no. 5, pp. 509–517, Nov. 1993.
- [3] C. T. Farley and D. P. Ferris, "Biomechanics of walking and running: Center of mass movements to muscle action," *Exercise and Sport Science Reviews*, vol. 26, pp. 253–283, 1998.
- [4] M. H. Raibert, *Legged Robots that balance*. MIT Press, 1986.
- [5] G. Zeglin, "The Bow Leg Hopping Robot," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, USA, Oct. 1999.
- [6] J. W. Hurst, J. Chestnutt, and A. Rizzi, "Design and Philosophy of the BiMASC, a Highly Dynamic Biped," in *IEEE Int. Conf. on Robotics and Automation*, Roma, Italy, Apr. 2007.
- [7] M. Ernst, H. Geyer, and R. Blickhan, "Spring-Legged Locomotion on Uneven Ground: a Control Approach to Keep the Running Speed Constant," in *12th International Conference on Climbing and Walking Robots (CLAWAR)*, 2009, pp. 639–644.
- [8] Ö. Arslan, U. Saranlı, and Ö. Morgül, "Reactive Footstep Planning for a Planar Spring Mass Hopper," in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, St. Louis, USA, Oct. 2009.
- [9] E. T. Whittaker, *A treatise on the analytical dynamics of particles and rigid bodies*, 4th ed. Cambridge University Press, New York, 1904.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] J. Schmitt and J. Clark, "Modeling posture-dependent leg actuation in sagittal plane locomotion," *Bioinspiration and Biomimetics*, vol. 4, pp. 1–17, 2009.
- [14] D. Dudek and R. J. Full, "Passive mechanical properties of legs from running insects," *Journal of Experimental Biology*, vol. 209, pp. 1502–1515, 2006.
- [15] 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.