Precision Robotic Leaping and Landing Using Stance-phase Balance

Justin K. Yim[1], B. Roodra P. Singh[2,3], Eric K. Wang[4], Roy Featherstone[2], and Ronald S. Fearing[1]

***Abstract*—Prior work has addressed control of continuous jumping using touchdown angle from flight, but greater precision can be obtained by directing individual leaps using liftoff angle from stance. We demonstrate targeted leaping as well as balanced landing on a narrow foot with a small, single leg hopping robot, Salto-1P. Accurate and reliable leaping and landing are achieved by the combination of stance-phase balance control based on angular momentum, a launch trajectory that stabilizes the robot at a desired launch angle, and an approximate expression for selecting touchdown angle before landing. Dynamic transitions between standing, hopping, and standing again are now possible in a robot with a narrow foot. We also present approximate bounds on acceptable velocity estimate and angle errors beyond which balanced landing is infeasible. Compared to a prior Spring Loaded Inverted Pendulum (SLIP)-like gait, the jump distance standard deviation is reduced from 9.2 cm to 1.6 cm for particular jumps, now enabling precise jumps to narrow targets.**

***Index Terms*—legged robots, dynamics, jumping**

I. Introduction

JUMPING can cross large distances and heights to clear gaps and obstacles, but it necessitates accurate control. During launch and landing, a locomotor must balance on its feet while quickly accelerating its body. Accurate launch velocity is required to jump to a small target, while accurate landing balance is required to remain on the target without tumbling. Exemplary demonstrations of these motions among animals include arboreal leaps between branches by squirrels, rapid traversals of sheer cliffs by mountain goats, and acrobatic jumps on balance beams by human gymnasts. Robots capable of similar high performance leaping and landing could better approach the mobility of animals in complex environments. In this work, we combine high performance balance control developed in [14] and [10] with high-power jumping to achieve two behaviors: precise leaping to targets and balanced landing as shown in Fig. 1. This is demonstrated on the jumping robot Salto-1P developed in [16] and [17].

Manuscript received: September 10, 2019; Revised: January 3, 2020; Accepted: February 8, 2020

This paper was recommended for publication by Editor Nikos Tsagarakis upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by Army Research Office Grant No. W911NF-18-1-0038.

[1]Justin K. Yim and Ronald S. Fearing are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, USA Justin K. Yim: yim@eecs.berkeley.edu

[2]B. Roodra P. Singh and Roy Featherstone are with the Department of Advanced Robotics, Istituto Italiano di Tecnologia, Genoa, Italy

[3]B. Roodra P. Singh is also with the Department of Information Engineering, University of Pisa, Italy

[4]Eric K. Wang is with the Department of Mechanical Engineering, University of California, Berkeley, USA

Digital Object Identifier (DOI): see top of this page.

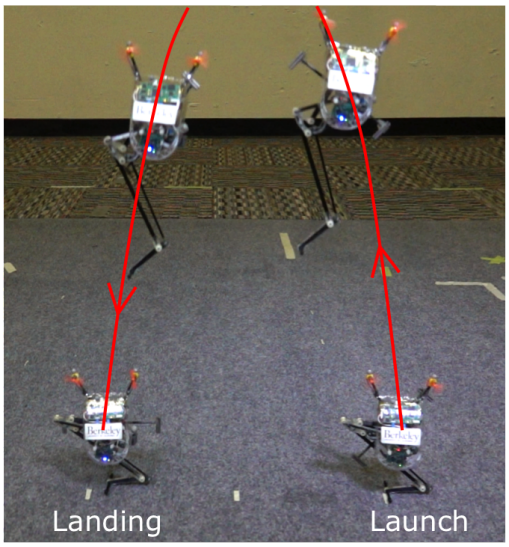


Fig. 1: Salto-1P leaps and lands.

A. Related Work

We organize our overview of related work into statically stable jumpers, dynamic runners, robots with many degrees of freedom, and platforms for investigating balance control.

Many robots aim their jumps from a statically stable initial posture [32], [8], [1], [39]. A similar approach uses steady running before launch to aim jumps [19]. These strategies can effectively direct extremely large jumps. However, to the best of the authors' knowledge, these robots do not attempt to control their landing posture unless by using additional multimodal aerodynamic appendages as in [21]. A righting maneuver may be required before another jump can be performed; furthermore, uncontrolled landing could cause a robot to bounce or tumble off of its target.

Running robots incorporate jumping into dynamic gaits. This motion is often described by the Spring Loaded Inverted Pendulum (SLIP) model [7]. Marc Raibert experimentally demonstrated impressive balanced SLIP-like hopping with simple heuristic control [26]. Many SLIP-like models and robots have followed. Boom-mounted robots have demonstrated precise step control [38], [33]. Deadbeat hopping over varying terrain is demonstrated in simulation by [34], [12], [2]. Deadbeat hopping can also be achieved using approximate solutions to the SLIP equations [27]. In previous work [36], aerial control sets touchdown angle and leg length to achieve desired takeoff velocities after one open-loop stance phase for Salto-1P. However, these works do not address the problem of landing and stopping.

Quadrupeds and hexapods with many degrees of freedom have demonstrated impressive acrobatic leaping and landing including hurdles over obstacles and flips [18], [24], [20]. However, these robots usually land on multiple contacts over a broad base of support and not a narrow area. Whole-body controllers, hybrid zero dynamics, and other optimization-based methods from [9] [31] [35] can execute complex maneuvers including leaping, landing, and gait transitions due to their algorithmic generality, but they require significant computation online or in pre-planning. On the other extreme, the severely under-actuated Acrobot can hop and land, but its motions are limited as shown in simulation by [5].

In walking robots, capture regions describe how to place a foot in order to arrest a robot's motion [25]. In this work, we derive a similar landing leg angle strategy to arrest a jumping robot and bring it close to a balanced posture.

After landing, the major challenge is balance on a small base of support. Balancing, as considered in control literature, is often regarded purely as a control exercise, ignoring the balancing ability of the plant. However [13] and [14] recently presented a different approach to balancing based on more thorough analysis of the physics of the plant.

Some interesting results were already available in the balancing of under-actuated robots such as Acrobots [30], [6] and the Pendubot [28] two decades ago; however, the performance was limited to swing-up control or tracking certain special trajectories while remaining balanced. Similar to the approach in [14], the controllers in [30] and [23] employed angular momentum as the state variables for a Reaction Wheel Pendulum (RWP) in [29] and for an output zeroing controller in [23]. In [29], the controller used feedback linearization of the dynamics of a RWP and then pole-placement for the resulting linear chain of integrators, while [23] relied on output zeroing of the angular momentum L that results in the robot being balanced. Another interesting balancing result based on the control of angular momentum for graphical simulations is given in [22]. In this paper, we present an angular momentum based leaning controller following from the high-performance balancing control in [14] and [10] combined with launch and landing strategies.

II. Models and Control Design

A. Motivation and Principles

We aim to produce large, accurate leaps and reliable landings on a narrow support. This is relevant to jumpers with one or two small feet or any jumper that must contact the ground only in a small region (on a ledge, for example).

While the robot is on the ground in stance phase, we consider its motion in two parts: leaning motion of the center of gravity (CG) as it rotates over its support, and radial motion of the CG towards and away from the support. We split control of these motions into *leaning control* and *leg control* respectively. The robot's ballistic flight trajectory always lies in a vertical plane and we assume that the robot confines its stance motions to this same plane.

During launch stance phase, leaning control and leg control follow a *leaning trajectory* so that the robot launches on a desirable flight path. In flight phase following liftoff, the *touchdown plan* sets the posture for landing. In landing stance phase after touchdown, leaning control balances the robot and leg control slows it (Fig. 2).

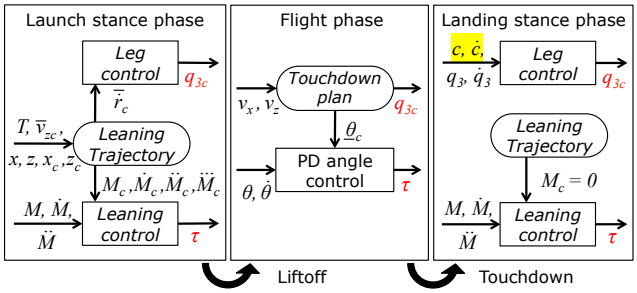


Fig. 2: Control block diagram for launch, flight, and landing.

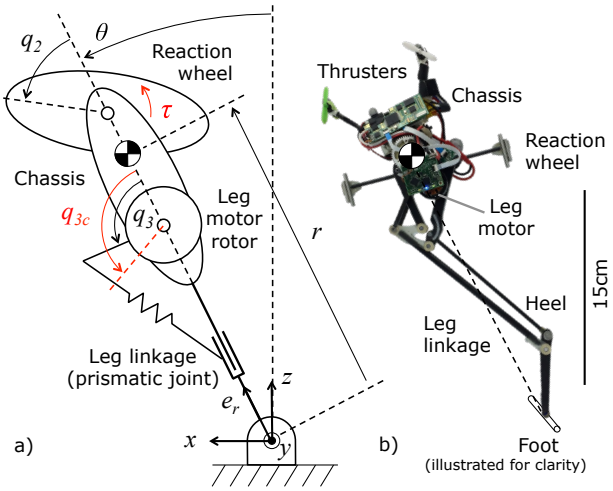


Fig. 3: a) Planar model of the robot in stance phase. The foot behaves like a pin joint when the robot is on the ground. b) Image of Salto-1P with its components labeled.

A jumping robot with a ballistic flight phase (i.e., one without large aerodynamic forces), has no control over its CG motion in the air, making launch conditions critical to accurate landing. In order to improve launch angle accuracy, we choose a launch condition with zero angular velocity of the CG about the support on liftoff and net zero angular momentum on liftoff. This limits the achievable launch angles, but partially decouples lean and leg control. In section II-G, we analyze the sensitivity of launch velocity to launch angle errors using this strategy and show that it is more accurate than SLIP-like running for similar leg angle errors.

To make a balanced landing on a narrow foot and avoid tumbling, the robot must touch down with acceptable angles and velocities. In section II-F, we derive approximate limits for balanced landing.

Both launch and landing depend critically on lean angle. This section focuses on lean angle, while development of leg control is detailed in section III.

B. Robot Model and Parameters

The robot Salto-1P consists mainly of three rigid bodies: the chassis, reaction wheel, and leg motor rotor with moments of inertia about the y axis of I_1, I_2, and I_3 respectively (Fig. 3).

Pin joints parallel to y join the chassis to the reaction wheel and leg motor through the latters' CGs so that their rotations, q_2 and q_3 respectively, do not move the CG. The straight-line leg linkage is considered to be a massless prismatic joint connecting the chassis and foot along axis e_r. We replaced Salto-1P's previous point foot with a bar parallel to y so that it behaves as a pin joint with angle θ as long as ground reaction forces remain inside the friction cone. The robot's total mass is m and its CG lies on e_r a distance r from the foot. $r_{\min}$ is the minimum length of r in full crouch and $r_{\max}$ is the maximum at full extension. Bars over ($\bar{}$) and under (_) variables denote values at liftoff and touchdown respectively.

The robot has two main actuation inputs: a torque τ at the q_2 joint and a command for the series-elastic leg motor. The leg motor is attached through a 25:1 gearbox to a torsional spring on the leg linkage input crank (gear, spring, and linkage depicted as lever arm, linear spring, and slider in Fig. 3). Proportional Derivative (PD) control of the leg motor follows a commanded angle q_{3c}. Additionally, two aerodynamic thrusters produce torques used only to stabilize the robot's motion to the $x-z$ plane in both flight and stance.

C. Leaning Plant Model

For leaning, we use the plant given in [14] based on the angular momentum L of the robot about its foot. Its complete dynamics is a chain of integrators that makes the controller design a simple pole-placement feedback problem. We also retain from [10] the use of scaled angular momentum $M = \frac{L}{mrg}$ instead of L where g is the acceleration due to gravity (a positive quantity). For Salto-1P, if r and q_3 are locked, it is equivalent to a RWP conveniently characterized by two parameters: the time constant of toppling of the entire robot considered as one rigid body T_t and the angular velocity gain of the reaction wheel G_ω as in [14] and [13] respectively:

$$T_t = \sqrt{\frac{mr^2 + I_1 + I_2 + I_3}{mrg}}, \quad G_\omega = -\frac{I_2}{mr^2 + I_1 + I_2 + I_3}$$

and the robot's rotational dynamics are described by the RWP equations of motion:

$$\begin{aligned} H_{11}\ddot{\theta} + H_{12}\ddot{q}_2 &= mrg\sin(\theta) \\ H_{21}\ddot{\theta} + H_{22}\ddot{q}_2 &= \tau \end{aligned} \tag{1}$$

where $H_{11} = mrg{T_t}^2$ and $H_{12} = H_{21} = H_{22} = -G_\omega H_{11}$. Another important quantity derived in [11] is the largest angle for which a RWP can recover a balanced position starting from rest. We will call this $\theta_{\max}$ ($\phi_{\max|\mathrm{I}}$ in [11]).

D. Leaning Control

The leaning control is responsible for the angular momentum and body angle of the robot during its launch and landing stance phases. Based on the theory of balancing given in [14] and experimentally verified on a RWP named Tippy in [10], a modified planar balance controller for the lean of the robot is presented in this subsection. This leaning controller tracks an angular momentum command instead of the position of the actuated joint as in [14], [10]. The state variables for this controller are chosen to be M, $\dot{M}$, $\ddot{M}$. They are calculated using only T_t and G_ω as follows:

$$M = {T_t}^2(\dot{\theta} - G_\omega\dot{q}_2), \quad \dot{M} = \theta, \quad \ddot{M} = \dot{\theta} \tag{2}$$

The leaning control law, similar to the controller in [14], can then be formulated based on full state feedback as:

$$\dddot{M} = k_{dd}\ddot{M} + k_d\dot{M} + k_m(M - u) \tag{3}$$

where u is the commanded angular momentum. The closed loop equation of motion for the control input (2) is:

$$\begin{bmatrix} \dddot{M} \\ \ddot{M} \\ \dot{M} \end{bmatrix} = \begin{bmatrix} k_{dd} & k_d & k_m \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \ddot{M} \\ \dot{M} \\ M \end{bmatrix} - \begin{bmatrix} k_m u \\ 0 \\ 0 \end{bmatrix}. \tag{4}$$

The feedback gains k_{dd}, k_d, k_m in (2) can now easily be determined using closed loop poles λ_i of the system.

$$\begin{aligned} k_{dd} &= \lambda_1 + \lambda_2 + \lambda_3 \\ k_d &= -(\lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_1\lambda_3) \\ k_m &= \lambda_1\lambda_2\lambda_3 \end{aligned}$$

The transfer function from the commanded angular momentum input to the output differs from that of the balance controller that tracks q_2 in [10]. Here the transfer function is:

$$\frac{M}{M_c} = \frac{-k_m(1 - \alpha_1 s + \alpha_2 s^2 - \alpha_3 s^3)}{(s^3 - k_{dd}s^2 - k_d s - k_m)} \tag{5}$$

where α_1, α_2 and α_3 are the feedforward gains. The zeros in the transfer function in (5) make u the combination of M_c and its first three derivatives:

$$u = M_c - \alpha_1\dot{M}_c + \alpha_2\ddot{M}_c - \alpha_3\dddot{M}_c$$

where the gains for the three zeros at μ_1, μ_2 and μ_3 are:

$$\alpha_3 = \frac{1}{\mu_1\mu_2\mu_3}, \quad \alpha_2 = \frac{\mu_1 + \mu_2 + \mu_3}{\mu_1\mu_2\mu_3}$$
$$\alpha_1 = \frac{\mu_1\mu_2 + \mu_2\mu_3 + \mu_1\mu_3}{\mu_1\mu_2\mu_3}.$$

TABLE I: Notation

Coordinates	**Symbol**	**Units**	
CG distance from foot	r	m	
lean angle	θ	rad	
reaction wheel angle	q_2	rad	
leg motor angle	q_3	rad	
Variables	**Symbol**	**Units**	
Ang. momentum about foot	L	N m s	
Scaled angular momentum	M	rad s	
Reaction wheel torque	τ	N m	
Parameters	**Symbol**	**Units**	**Value (crouch)**
Total mass	m	kg	0.111
Chassis MoI	I_1	kg m^2	1.2×10^{-4}
Reaction wheel MoI	I_2	kg m^2	3.3×10^{-5}
Leg motor rotor MoI	I_3	kg m^2	5×10^{-7}
Min (crouched) r	r_{min}	m	0.090
Max (extended) r	r_{max}	m	0.234
Time constant of toppling	T_t	s	0.10
q_2 angular velocity gain	G_ω	—	-0.032
q_2 stall torque	τ_{stall}	N m	0.045
q_2 free-running speed	ω_{free}	rad/s	120
Max recovery angle [11]	θ_{max}	rad	0.218

Even though the placement of zeros is not necessary for tracking the angular momentum command, it aids in the closed loop control action being equivalent to a linear filter due to stable pole-zero cancellation. In launch, the M_c trajectory is provided by the launch trajectory plan in section II-E, while on landing, M_c and its derivatives are set to 0 to command the robot to balance upright.

Although the output of the leaning controller is $\dddot{M}$ (where $\dddot{M} = \ddot{\theta}$), this output can easily be converted to a torque command τ for the actuated joint using the robot's equation of motion given by (1):

$$\ddot{q}_{2c} = (mrg\sin(\theta) - H_{11}\dddot{M})/H_{12}$$
$$\tau = H_{21}\dddot{M} + H_{22}\ddot{q}_{2c}\ .$$

As mentioned in [3], [4], the balance controllers are sensitive to the estimate of vertical, and [10] introduced a balance offset observer to correct this estimate. These offsets can arise due to drift in sensors such as IMUs. During stance phase, the balance offset observer estimates and compensates for sensor drift assuming it varies slowly. The estimated drift θ_o is subtracted from the reading of θ from the sensor, $\hat{\theta}$. Hence the $\dot{M}$ state in (3) uses an updated value of $\theta = \hat{\theta} - \theta_o$. The balance offset estimator is disabled just prior to the robot's brief launch motion since the robot occasionally bumps its leg linkage on the ground during aggressive launch leans; this would perturb the offset estimator were it active.

E. Leaning Trajectory

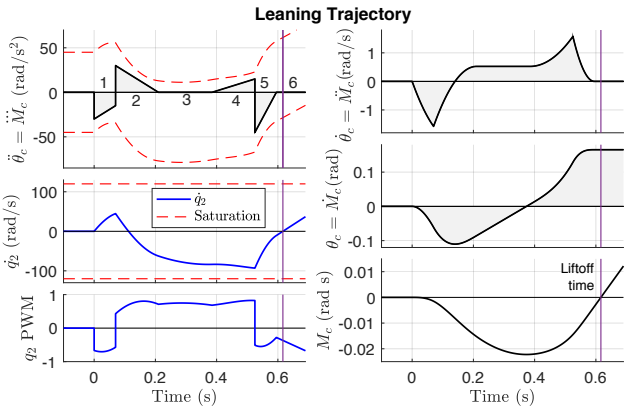
The robot must achieve liftoff vertical and horizontal CG velocities $\bar{v}_z$ and $\bar{v}_x$ respectively that will launch it on a ballistic parabola from which it can land on its target.

We choose a stance trajectory in which θ leans to a desired angle $\bar{\theta}_c$ while the leg remains fixed at $r = r_{\min}$, after which θ remains fixed at $\bar{\theta}_c$ while r rapidly extends to desired liftoff velocity $\bar{\dot{r}}_c$. Thus $\bar{v}_x = \bar{\dot{r}}\sin(\bar{\theta})$ and $\bar{v}_z = \bar{\dot{r}}\cos(\bar{\theta})$. This trajectory decouples lean and leg control by sequencing them one after the other.

We would like this trajectory to achieve the following qualities: 1) $\bar{\theta} = \bar{\theta}_c$, 2) $\dot{\bar{\theta}} = \bar{\dot{q}}_2 = 0$, 3) $\bar{L} = 0$, and 4) the q_2 reaction wheel motor avoids saturation. It can be shown that any trajectory with these qualities is limited to angles $|\theta| \leq |\theta_{\max}|$. We hand-designed a $\theta_c(t) = \dot{M}_c(t)$ launch trajectory as a sequence of piecewise cubic functions (segments of constant $\dddot{M}_c$). It is parameterized by an angular acceleration scale a in rad/s^2 and a time scale T in seconds (shown in Fig. 4). While many trajectories could achieve the four desired qualities, this trajectory was selected for its simple analytical solution and range of valid parameters.

This trajectory dwells at $\bar{\theta}_c = \frac{9aT^2}{8}$ rad starting at $8.5T$ s. Liftoff should occur when $M = \dot{q}_2 = 0$, which is slightly later at $\frac{317}{36}T$ s. Due to the lean angle dwell, mistimed liftoff will perturb $\bar{L}$ and $\bar{\dot{q}}_2$ away from zero, but $\bar{\theta}$ will remain at $\bar{\theta}_c$.

To launch to a target, a jump planner takes as input the current foot location (x, z), desired foot location (x_c, z_c), desired vertical velocity $\bar{v}_{zc}$, and trajectory time scale T. The planner solves for the required lean angle and value of a using



Segment	Time	$\dddot{M}_c$ initial	$\dddot{M}_c$
1	$0 \leq t < T$	$-a$	$0.5a/T$
2	$T \leq t < 3T$	a	$-0.5a/T$
3	$3T \leq t < 5.5T$	0	0
4	$5.5T \leq t < 7.5T$	0	$0.25a/T$
5	$7.5T \leq t < 8.5T$	$-1.5a$	$1.5a/T$
6	$8.5T \leq t$	0	0

Fig. 4: Leaning trajectory for launch with parameters a = 30 rad/s^2 and T = 0.07 s. Intended liftoff time is indicated by the purple line.

small angle approximation for the position of the CG on liftoff and assumes $\bar{r}$ is close to $r_{\max}$:

$$\bar{v}_{xc} = \frac{(x_c - x)g\bar{v}_{zc}}{2r_{\max}g + \bar{v}_{zc}\sqrt{\bar{v}_{zc}^2 - 2(z_c - z)g} + \bar{v}_{zc}^2} \tag{6}$$

$$\bar{\theta}_c = \operatorname{atan}\left(\frac{\bar{v}_{xc}}{\bar{v}_{zc}}\right) \tag{7}$$

$$a = \frac{8}{9T^2}\left(\bar{\theta}_c\right) \tag{8}$$

F. Touchdown Plan

Flight control sets the initial conditions for landing. Its two components, touchdown angle and leg length, must be adjusted for the velocity at touchdown since an incorrect touchdown could be unrecoverable for the leaning controller.

1) Touchdown leg length: After the robot reaches apex, its leg should extend as it falls. Ideally, at any velocity, $\underline{r}$ should be just long enough so that r quickly reaches $r_{\min}$ without violently striking the end of its leg stroke. This maximizes the robot's physical ability to balance [13]. This relationship was found experimentally in Salto-1P and is described with other leg control details in section III-A.

2) Touchdown angle: Appropriate selection of touchdown leg angle $\underline{\theta}$ is critical to successful landing. Impact with the ground sets the initial angular velocity of the landing phase. An ideal post-impact angular velocity would carry the robot close to vertical with a minimum of control effort in order to maximize the recovery margin. This is equivalent to zeroing the effective offset angle from [11] after impact:

$$\underline{\theta}^+_{\text{eff}} = \underline{\theta} + \underline{\dot{\theta}}^+ T_t = 0 \tag{9}$$

where $^+$ denotes a value immediately after transition, and T_t is the time constant of toppling. The conservation of angular

momentum for touchdown is:

$$\left(m\underline{r}^2 + I_1\right)\dot{\underline{\theta}}^+ = -m\underline{r}\left(\underline{v}_z^- \sin(\underline{\theta}) - \underline{v}_x^- \cos(\underline{\theta})\right) + I_1\dot{\underline{\theta}}^- \tag{10}$$

where $^-$ denotes a value immediately before transition.

These expressions can be simplified with several assumptions: 1) We assume $\dot{\underline{\theta}}^-$ is negligible due to flight-phase attitude control. 2) We assume $I_1 << mr^2$ and neglect I_1. This assumption is good for robots with long, light limbs (often advantageous for jumping). 3) We make a small angle approximation.

With the above assumptions, (10) simplifies to:

$$\dot{\underline{\theta}}^+ = \frac{-\underline{\theta}\underline{v}_z^- + \underline{v}_x^-}{\underline{r}} \tag{11}$$

Solving (9) and (11) for $\underline{\theta}$ produces the desired angle $\underline{\theta}_c$:

$$\underline{\theta}_c = -\frac{T_t\underline{v}_x^-}{\underline{r} - T_t\underline{v}_z^-} \tag{12}$$

For simplicity, we set $\underline{r} = r_{\min}$. This final approximation creates a $\dot{\underline{\theta}}^+$ with magnitude slightly too small to zero $\underline{\theta}_{\text{eff}}^+$.

(12) can also be used to find approximate bounds on $\underline{\theta}$ outside of which the robot cannot balance. Combining (9) and (11), then substituting $\pm\theta_{\max}$ for $\underline{\theta}_{\text{eff}}^+$ and solving for $\underline{\theta}$ yields the maximum acceptable error $\underline{\theta}_e$:

$$\underline{\theta}_e = \pm\frac{\underline{r}\theta_{\max}}{\underline{r} - T_t\underline{v}_z^-} \tag{13}$$

This is the approximate maximum allowable touchdown angle error due to the combination of estimator error and flight-phase attitude control error. For Salto-1P with $\underline{r} = r_{\min}$ and $\underline{v}_z^- = -4$ m/s, $\underline{\theta}_e = \pm 0.040$ rad (only $\pm 2.3°$).

Velocity estimate error will also contribute to touchdown angle error through (12). Equating (12) and (13) we solve for the maximum acceptable horizontal velocity error $\underline{v}_{\text{e}}^-$:

$$\underline{v}_{\text{e}}^- = \frac{\underline{r}\theta_{\max}}{T_t} \tag{14}$$

At $\underline{r} = r_{\min}$ and with otherwise perfect state estimation and control, Salto-1P's horizontal velocity estimate error must be less than than approximately 0.20 m/s to avoid falling over.

G. Precision

Both stance-phase control presented here and prior flight-phase control use leg angle to control jump velocity: liftoff angle $\bar{\theta}$ for stance-phase control and touchdown angle $\underline{\theta}$ for flight-phase control. Consequently, errors in angle control will propagate to velocity errors with consequences for the precision of targeted jumps.

Using small angle approximation, the gain from variation in $\bar{\theta}$ to horizontal velocity $\bar{v}_x$ using stance control is simply

$$\frac{\partial\bar{v}_x}{\partial\bar{\theta}} = \bar{v}_z \tag{15}$$

which ranged from 2.0 to 4.0 m/s per rad in Results IV-A.

We derive a similar approximate expression for the sensitivity of flight-phase control from [36] to touchdown angle for comparison (Fig. 5). Many works provide approximate solutions to the SLIP dynamics, such as [15], [27]. As in [15], by setting $\bar{r} = \underline{r}$, neglecting gravity, and using conservation of energy, only the angle change from touchdown $\underline{\theta}$ to liftoff $\bar{\theta}$ need be solved for. As a simple underestimate, we approximate $\bar{\theta} - \underline{\theta} = \dot{\underline{\theta}}^+ t_s$ with stance time t_s (0.07 s for Salto-1P). Together with (11) and a small angle approximation, $\bar{v}_x$ as a function of touchdown conditions becomes:

$$\bar{v}_x = \underline{v}_x^- - \underline{\theta}\underline{v}_z^- - (\underline{v}_z^- + \underline{\theta}\underline{v}_x^-)\left(\underline{\theta} + \frac{t_s(\underline{v}_x^- - \underline{\theta}\underline{v}_z^-)}{\underline{r}}\right) \tag{16}$$

Taking the partial derivative with respect to $\underline{\theta}$ and evaluating for hopping in place at $\bar{v}_x = \underline{v}_x^- = \bar{\theta} = \underline{\theta} = 0$ produces:

$$\frac{\partial\bar{v}_x}{\partial\underline{\theta}} = -2\underline{v}_z^- + \frac{t_s}{r}(\underline{v}_z^-)^2 \tag{17}$$

Since $\underline{v}_z^-$ is negative for usual running and $\underline{v}_z^- = -\bar{v}_z$ at this operating point, stance-phase control $\bar{v}_x$ sensitivity to $\bar{\theta}$ error in (15) is less than half the flight-phase control $\bar{v}_x$ sensitivity to $\underline{\theta}$ error in (17). Therefore, for small θ and similar angle errors and jump heights, stance-phase control should achieve more precise jumps than flight-phase control.

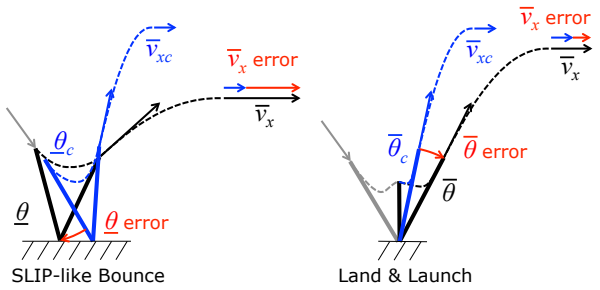


Fig. 5: Comparison of angle error sensitivity of flight-phase control in [36] and stance-phase control in this work.

III. Experimental Setup

The testbed for launching and landing is Salto-1P, a monopedal robot with maximum body length 0.313 m and parameters given in Table I. Salto-1P uses onboard encoders and gyroscopes to estimate q_2, q_3, r, and θ. It also uses them to estimate liftoff velocity as described in [37] with a horizontal error standard deviation of about 0.1 m/s, about half of the II-F error limit. To evaluate the reliability of the leaning control, leg control, and touchdown plan with less disruption by estimator noise, we augment velocity estimation with motion capture. Velocity measurements and lean trajectory commands streamed at 100Hz from a ground station computer radio. Salto-1P's onboard DsPIC33FJ128MC706A microcontroller computed estimation, leaning control, leg control, and touchdown plan at 500Hz.

A. Series-elastic Leg Control

CG motion along r is equivalent to the motion of a mass on a linear rail. However, Salto-1P uses nonlinear series-elastic power modulating leg actuation [16]. We control this nonlinear leg actuation with an energy-based leg controller.

During launch, leg control accelerates the robot to the desired radial velocity at liftoff, $\dot{\bar{r}}_c$, by rotating the leg motor

a fixed angle, q_{3c}. Approximating Salto-1P's torsional spring as linear, q_{3c} and $\dot{\bar{r}}_c$ are related by energy:

$$\frac{1}{2}k\left(\frac{1}{G}(q_{3c}-q_o)\right)^2=\frac{1}{2}m\dot{\bar{r}}_c^2 \tag{18}$$

where $G=25$ is the transmission ratio and k and q_o are effective spring stiffness parameters considering transmission losses and spring nonlinearity. Solving for q_{3c} yields:

$$q_{3c}=G\sqrt{\frac{m}{k}}\dot{\bar{r}}_c+q_o \tag{19}$$

For Salto-1P's experimentally measured parameters: $q_{3c}=17\left(\frac{\text{rad s}}{\text{m}}\right)\dot{\bar{r}}_c+18.5$ rad.

Salto-1P lifts off approximately 0.14 s after activation of its leg motor, so the leg motor is activated slightly before intended liftoff time at $\frac{317}{36}T-0.14$ s.

In flight phase after apex, leg length is set by

$$q_{3c}=-10v_z+25 \tag{20}$$

increasing $\underline{r}$ as $|v_z|$ increases.

During landing, Salto-1P uses closed-loop force control of its series-elastic leg actuator to emulate a damper with damping coefficient 1.5 N s m^{-1} so that jump energy is smoothly removed as r rapidly compresses to $r_{\min}$. This coefficient and the force control gains were found experimentally.

B. Leaning and Leg Coupling

Leaning and leg control are coupled by Coriolis and centrifugal fictitious forces as well as gravity. We mitigate these couplings by selection of launch trajectory, but the leg motor action also perturbs leaning control.

When active, the leg motor applies torque to the chassis at q_3. For negative $\bar{\theta}_c$, this torque acts in the same direction as τ and generates little deviation from the leaning trajectory. However, it saturates the q_2 motor and generates a negative deviation of $\bar{\theta}$ for positive $\bar{\theta}_c$. To reduce this disturbance, the leg motor pre-winds 30 rad and $\bar{\theta}_c$ is increased by $\Delta\theta$ to an *adjusted lean angle* for forward jumps:

$$\Delta\theta=-\text{sign}(\bar{v}_{xc})G_{\omega 3}(q_{3c}-30) \tag{21}$$

$$G_{\omega 3}=-\frac{I_3}{mr_{\min}^2+I_1+I_2+I_3}\approx -5\times 10^{-4} \tag{22}$$

Since touchdown q_3 rotation for leg retraction perturbs θ forwards during landing (as it does backwards during launch), the planned touchdown angle $\underline{\theta}_c$ is offset by $-1°$.

In addition, extension of r changes T_t and G_ω, requiring a change in leaning controller gains. On Salto-1P, we chose $\lambda_1=\lambda_2=\lambda_3=\mu_1=\mu_2=\mu_3=-12$ when $r=r_{\min}$ and -9 when r is at least partially extended.

IV. Results

A. Launch and Landing

We tested Salto-1P's ability to launch at a grid of desired velocities and then land, shown in Fig. 6. Commanded launch angles ranged from -75% to +75% of $\theta_{max}=0.218$ rad: $\bar{\theta}_c=0,\pm0.110$, and ±0.165 rad ($T=0.07$ s and $a=0,\pm20,\pm30$ rad/s^2). The action of the leg motor reduced the adjusted lean angles by $5\times10^{-4}(q_3-30)$ rad for positive lean angles due to reaction wheel saturation. Commanded leg velocities $\dot{\bar{r}}_c$ were 1.88, 2.88, and 3.88 m/s. Each combination of launch angle and leg velocity was tested four times for a total of 60 trials.

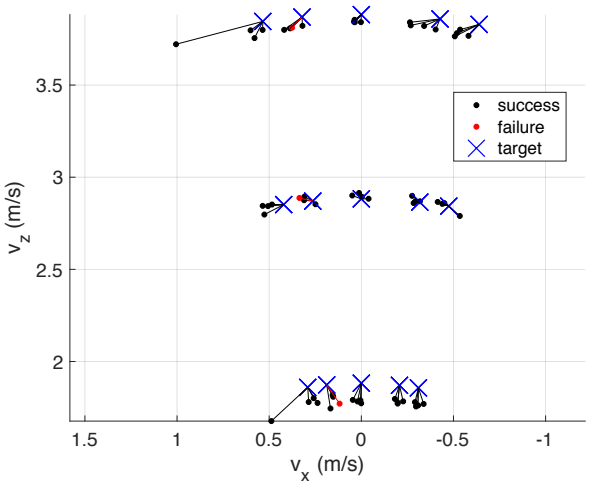


Fig. 6: Launch and landing tests at a grid of liftoff velocities. 95% success in 60 trials (failures shown in red).

For these launches, Salto-1P achieved launch angle errors with a standard deviation of 0.023 rad (1.3 degrees) across this range. The horizontal and vertical velocity error means were 0.041 m/s and -0.048 m/s respectively and the standard deviations were 0.079 m/s and 0.047 m/s respectively. When fully crouched, Salto-1P's lowest leg link makes a 0.15 rad angle with the horizontal from the foot to the joint labeled "heel" in Fig. 3. Launches at -0.165 rad caused the heel to bump the ground. This may explain the consistent positive $\bar{v}_x$ error for trials at -0.165 rad. Salto-1P landed 57 out of 60 trials, falling over forwards once each at 0.110 rad and 1.88 m/s, 0.110 rad and 2.88 m/s, and 0.110 rad and 3.88 m/s. In five trials Salto-1P did not fall over, but came to rest on its heel rather than on the point of its foot alone.

B. Launch Accuracy

To evaluate launch precision, we tested a moderately large jump with trajectory parameters $a=30$ rad/s^2, $T=0.07$ s, and $\dot{\bar{r}}_c=3.38$ m/s, corresponding to an unadjusted launch angle of 0.166 rad (0.147 rad adjusted angle). The apex is 0.571 m above liftoff (just under two bodylengths) and the horizontal displacement on flat ground is 0.326 m (just over one bodylength). The leaning trajectories are shown in Fig. 7 and the resulting ballistic flight paths are shown in Fig. 8.

TABLE II: Jump Capabilities

	Previous [36]	**Results IV-B**	**Results IV-A**
Can land and stop	no	yes	
Largest tested $\lvert\bar{v}_{xc}\rvert$ (m/s)	1.68	0.64	0.48
Tested $\bar{v}_{zc}$ range (m/s)	2.4 - 3.9	1.9 - 3.9	3.3
θ error STD (rad)	0.010	0.019	0.010
Sensitivity $\frac{\partial\bar{v}_x}{\partial\theta}$ (m/s per rad)	7 - 17	2 - 4	3.3
x error STD (cm)	9.2	5.6	1.6

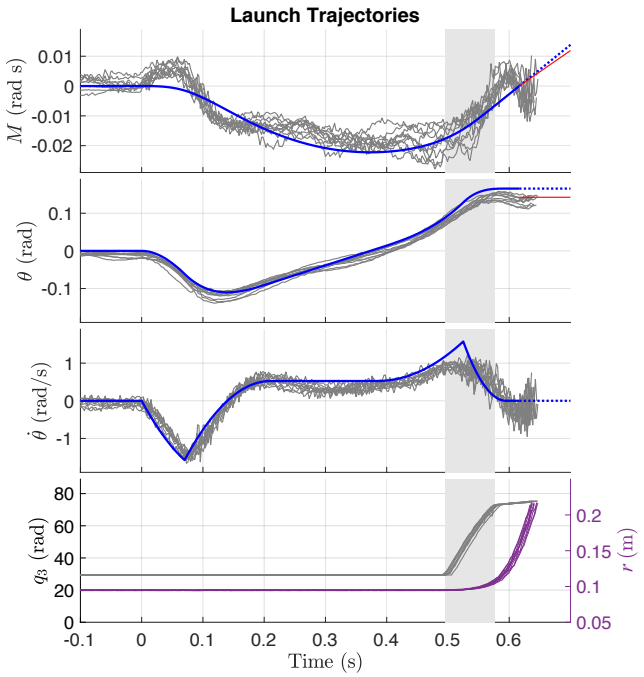


Fig. 7: 10 launch trajectories to unadjusted launch angle 0.166 rad, adjusted angle 0.147 rad (red). Reference in blue, trajectories in grey. Duration of leg motor rotation shown as light grey.

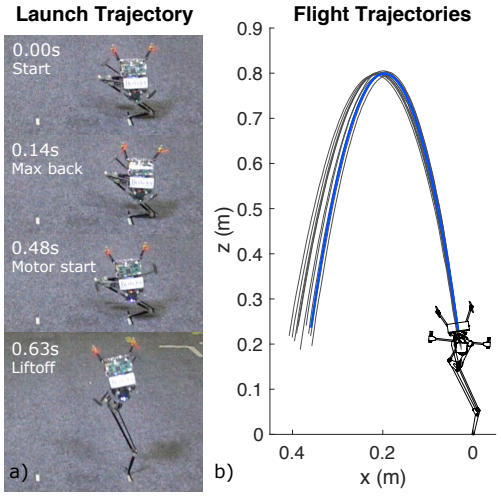


Fig. 8: 10 jumps testing accuracy at nominal distance 32.6 cm; achieved mean 35.1 cm, standard deviation 1.6 cm. a) Launch stance trajectory. b) Resulting flight trajectories (desired in blue).

The achieved $\bar{\theta}$ angles had a mean of 0.137 rad and standard deviation of 0.010 rad. The achieved launches had standard deviations in $\bar{v}_z$ of 0.010 m/s and $\bar{v}_x$ of 0.023 m/s. The mean jump distance was 0.351 m and its standard deviation was 0.016 m. Table II compares this performance to flight-phase control in [36] which could not land and stop.

Attitude error was similar for both stance-phase control and flight-phase control; liftoff attitude error of the balance controller in this work and touchdown attitude error of the aerial attitude control in [36] both had standard deviations of 0.010 rad. Therefore, lower sensitivity to attitude error may explain the higher accuracy of stance control.

The stance-phase control achieves jump precision between 1.6 and 5.8 times better than the flight-phase control, in reasonable agreement with the approximate relationships described in section II-G. The launch angle error for the full range of jumps in IV-A was worse than for the 10 moderate jumps in IV-B due mostly to lower accuracy at negative angles and two outliers at $a = 30$.

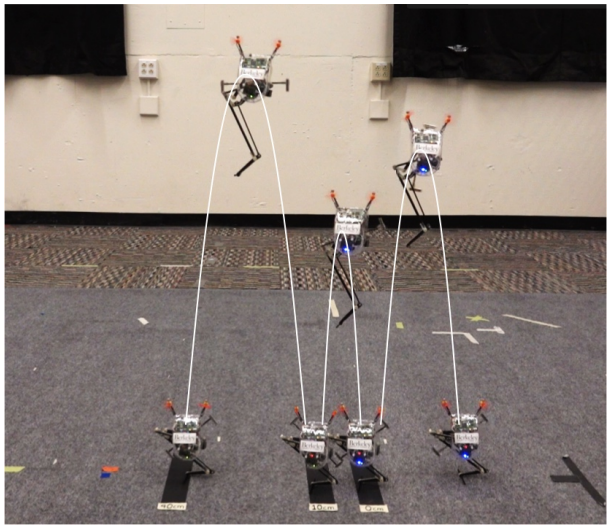


Fig. 9: Salto-1P leaps to and lands on consecutive narrow targets marked on the floor.

C. Multiple jumps to targets

Chaining together consecutive launches and landings, Salto-1P can land on smaller targets than was possible using flight-phase control in [36]. In Fig 9, the ground station computer uses motion capture to track Salto-1P and commands liftoff velocities that direct Salto-1P to points at 0 cm, 10 cm, and 40 cm. Starting at -23.4 cm, Salto-1P jumped to 0.8 cm, 11.0 cm, and 38.0 cm for errors of 0.8 cm, 1.0 cm, and -2.0 cm.

V. Conclusion

We demonstrate accurately targeted jumping and balanced landing on a narrow support. First, this work demonstrates higher precision jumping to a target than that achieved in [36]. The higher precision is likely due to the lower sensitivity to angle error associated with stance phase leaning control of launch compared to flight-phase attitude control of SLIP-like bouncing.

Second, we demonstrate balanced landing on a narrow support and present approximate limits on touchdown angle error and velocity estimate error in which balanced landing is possible. The tight error limits reveal why it is difficult to land a jump like a gymnast on a small base of support. This landing ability provides a transition from running to standing still and allows perching on small footholds. High-performance leaning control is critical to both accurate jumping and balanced landing since both depend on control of the robot's angle.

There are several areas for future improvement. Derivations of touchdown angle and balance limits assume small angle approximation and ignore leg inertia which make them inexact. Salto-1P can launch and land using only onboard processing and sensing, but it is less reliable without motion capture due

to the tight error limits. Instead of a line foot, Salto-1P can also land on a point foot, but its aerodynamic thrusters struggle since they were sized for aerial reorientation. More powerful roll actuation and improved velocity sensing can enable fully-autonomous jumping and stopping on more varied terrains.

There are also many possible future extensions to this work. The presented control can jump up or down ledges but this was not demonstrated for brevity. The selected leaning trajectory cannot exceed the maximum recovery angle (12.5° for Salto-1P). Other strategies could tilt and jump farther by forgoing zero angular velocity on liftoff with a potential tradeoff between accuracy and distance. Integrating leaning control with earlier flight-phase hopping control and coordinating out-of-plane motions in stance could produce 3D motions faster than stance-phase launching and more accurate than SLIP-like bouncing. Investigations of difficult surfaces with compliance and sliding can help expand operation out of the lab.

REFERENCES

[1] E. Ackerman, "Boston Dynamics Sand Flea robot demonstrates astonishing jumping skills," 2012. [Online]. Available: https://spectrum.ieee.org/automaton/robotics/military-robots/boston-dynamics-sand-flea-demonstrates-astonishing-jumping-skills

[2] A. Agrawal and K. Sreenath, "Bipedal robotic running on stochastic discrete terrain," *2019 18th European Control Conference, ECC 2019*, pp. 3564–3570, 2019.

[3] M. Azad, "Balancing and hopping motion control algorithms for an under-actuated robot," Ph.D. dissertation, Australian National University, 2014.

[4] M. Azad and R. Featherstone, "Angular momentum based balance controller for an under-actuated planar robot," *Autonomous Robots*, vol. 40, no. 1, pp. 93–107, 2016.

[5] M. Berkemeier and R. Fearing, "Sliding and hopping gaits for the underactuated Acrobot," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 629–634, 1998.

[6] ——, "Tracking fast inverted trajectories of the underactuated Acrobot," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 4, pp. 740–750, 1999.

[7] R. Blickhan, "The spring-mass model for running and hopping," *Journal of Biomechanics*, vol. 22, no. 1112, pp. 1217–1227, 1989.

[8] J. Burdick and P. Fiorini, "Minimalist jumping robots for celestial exploration," *The International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 653–674, 2003.

[9] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," *IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-February, pp. 295–302, 2015.

[10] J. J. M. Driessen, A. E. Gkikakis, R. Featherstone, and B. R. P. Singh, "Experimental demonstration of high-performance robotic balancing," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 9459–9465.

[11] J. J. Driessen, R. Featherstone, and A. E. Gkikakis, "An actuator design criterion to maximize physical balance recovery," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3829–3836.

[12] J. Englsberger, P. Kozlowski, C. Ott, and A. Albu-Schaffer, "Biologically inspired deadbeat control for running: from human analysis to humanoid control and back," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 854–867, 2016.

[13] R. Featherstone, "Quantitative measures of a robot's physical ability to balance," *International Journal of Robotics Research*, vol. 35, no. 14, pp. 1681–1696, 2016.

[14] ——, "A simple model of balancing in the plane and a simple preview balance controller," *International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1489–1507, 2017.

[15] H. Geyer, A. Seyfarth, and R. Blickhan, "Spring-mass running: simple approximate solution and application to gait stability," *Journal of Theoretical Biology*, vol. 232, no. 3, pp. 315–328, 2005.

[16] D. W. Haldane, M. M. Plecnik, J. K. Yim, and R. S. Fearing, "Robotic vertical jumping agility via series-elastic power modulation," *Science Robotics*, vol. 2048, no. 1, p. eaag2048, 2016.

[17] D. W. Haldane, J. K. Yim, and R. S. Fearing, "Repetitive extreme-acceleration (14-g) spatial jumping with Salto-1P," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, sep 2017, pp. 3345–3351.

[18] A. M. Johnson and D. E. Koditschek, "Toward a vocabulary of legged leaping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 2568–2575.

[19] G.-P. Jung, C. S. Casarez, S.-p. Jung, R. S. Fearing, and K.-J. Cho, "An integrated jumping-crawling robot using height-adjustable jumping module," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 6–11.

[20] B. Katz, J. D. Carlo, and S. Kim, "Mini Cheetah : a platform for pushing the limits of dynamic quadruped control," in *IEEE Int. Conf. on Robotics and Automation*, 2019, pp. 6295–6301.

[21] M. Kovač, Wassim-Hraiz, O. Fauria, J. C. Zufferey, and D. Floreano, "The EPFL jumpglider: a hybrid jumping and gliding robot with rigid or folding wings," in *IEEE International Conference on Robotics and Biomimetics, ROBIO 2011*, 2011, pp. 1503–1508.

[22] Macchietto and A. Patrick, "Momentum-based balance control for simulated characters," M.S., University of California, Riverside, 2008.

[23] N. Miyashita, M. Kishikawa, and M. Yamakita, "3D motion control of 2 links (5 D.O.F.) underactuated manipulator named AcroBOX," in *American Control Conference*, 2006, pp. 5614–5619.

[24] H. W. Park, P. M. Wensing, and S. Kim, "Online planning for autonomous running jumps over obstacles in high-speed quadrupeds," *Robotics: Science and Systems*, vol. 11, 2015.

[25] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: a step toward humanoid push recovery," *IEEE-RAS International Conference on Humanoid Robots, HUMANOIDS*, pp. 200–207, 2006.

[26] M. H. Raibert and J. H. B. Brown, "Experiments in balance with a 3D one-legged hopping machine," *Journal of Dynamic Systems, Measurement, and Control*, vol. 106, no. 1, pp. 75–81, 1984.

[27] U. Saranli, Ö. Arslan, M. M. Ankarali, and Ö. Morgül, "Approximate analytic solutions to non-symmetric stance trajectories of the passive Spring-Loaded Inverted Pendulum with damping," *Nonlinear Dynamics*, vol. 62, no. 4, pp. 729–742, 2010.

[28] M. W. Spong and D. J. Block, "The Pendubot: a mechatronic system," in *Proceedings of the 34th Conference on Decision & Control New Orleans, LA*, 1995, pp. 555–556.

[29] M. W. Spong, P. Corke, and R. Lozano, "Nonlinear control of the Reaction Wheel Pendulum," *Automatica*, vol. 37, no. 11, pp. 1845–1851, 2001.

[30] M. Spong, "Swing up control of the Acrobot," in *IEEE International Conference on Robotics and Automation*, 1994, pp. 2356–2361.

[31] K. Sreenath, H. W. Park, I. Poulakakis, and J. W. Grizzle, "Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on MABEL," *International Journal of Robotics Research*, vol. 32, no. 3, pp. 324–345, 2013.

[32] S. A. Stoeter, P. E. Rybski, M. Gini, and N. Papanikolopoulos, "Autonomous stair-hopping with Scout robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, no. October, 2002, pp. 721–726.

[33] P. Terry, G. Piovan, and K. Byl, "Towards precise control of hoppers: Using high order partial feedback linearization to control the hopping robot FRANK," in *IEEE 55th Conference on Decision and Control, CDC 2016*, 2016, pp. 6669–6675.

[34] A. Wu and H. Geyer, "The 3-D spring-mass model reveals a time-based deadbeat control for highly robust running and steering in uncertain environments," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1114–1124, 2013.

[35] X. Xiong and A. D. Ames, "Bipedal hopping: reduced-order model embedding via optimization-based control," *IEEE International Conference on Intelligent Robots and Systems*, pp. 3821–3828, 2018.

[36] J. K. Yim and R. S. Fearing, "Precision jumping limits from flight-phase control in Salto-1P," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2229–2236.

[37] J. K. Yim, E. K. Wang, and R. S. Fearing, "Drift-free roll and pitch estimation for high-acceleration hopping," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 8986–8992.

[38] G. Zeglin, "The bow leg hopping robot," Ph.D. dissertation, Carnegie Mellon University, 1999.

[39] J. Zhao, J. Xu, B. Gao, N. Xi, F. J. Cintron, M. W. Mutka, and L. Xiao, "MSU jumper: a single-motor-actuated miniature steerable jumping robot," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 602–614, 2013.