Title
Hopping Control and Estimation for a High-performance Monopedal Robot, Salto-1P

Author
Yim, Justin

Publication Date
2020

Peer reviewed|Thesis/dissertation

Hopping Control and Estimation for a High-performance Monopedal Robot, Salto-1P

by

Justin Yim

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering – Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ronald S. Fearing, Chair
Professor Robert J. Full
Professor Koushil Sreenath
Professor Ruzena Bajcsy

Spring 2020

Hopping Control and Estimation for a High-performance Monopedal Robot, Salto-1P

Abstract

Hopping Control and Estimation for a High-performance Monopedal Robot, Salto-1P

by

Justin Yim

Doctor of Philosophy in Engineering – Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Ronald S. Fearing, Chair

Jumping is an exciting locomotion mode that can enable small ground-based robots to maneuver around large obstacles and gaps. A high-power jumping robot can rapidly traverse obstacles, but the resulting fast and forceful stance phases are challenging for control and estimation. This dissertation presents control and estimation for precise hopping, operation using only onboard sensors, and precise leaps and balanced landings. These algorithms place an emphasis on simple models of motion to gain insights about jumping dynamics and to derive simple programs that can run on a small robot's computationally-limited onboard processor.

Control and estimation algorithms are experimentally demonstrated with Salto-1P, a small 0.1 kg robot with a 0.15 m leg exhibiting a 1.83 m/s vertical jumping agility, the highest of any untethered electrically actuated robot when introduced. First, with precise foot placement, Salto-1P climbs and descends obstacles higher than its bodylength without missing a step or crashing. Second, onboard estimation of attitude and jumping velocity enable Salto-1P to operate without offboard sensing or processing and run fully autonomously or accept human guidance to hop outdoors. Finally, using high-performance balance control, Salto-1P demonstrates precisely targeted leaps and balanced landings to jump to and stand on narrow targets.

To my parents and grandparents, for raising me with a love of invention, science, and engineering.
And to young inventors, scientists, and engineers. May your pursuits bring you excitement and joy.

Contents

List of Figures

List of Tables

Acknowledgments

My dissertation work and experience have been enriched by the support, discourse, and friendship of many people.

First, I would like to thank my advisor, Ron Fearing, for his support and guidance throughout my PhD studies. He has been an invaluable mentor to me. I would also like to thank my qualifying exam and dissertation committee, Ruzena Bajcsy, Bob Full, and Koushil Sreenath, for their feedback and support. An extra thank you to Bob Full for his instruction, tips, and insight in class and during our collaboration.

The ideas and conclusions presented in this dissertation are the results of work by myself with many colleagues. Their discussion and collaboration have brought me great joy and I am proud and delighted to have had the chance to work with them. Thank you to Duncan Haldane, Marc Plecnik, Eric Wang, Hersh Sanghvi, Roodra Singh, and Roy Featherstone. This work would not exist in its present form without you. Particular thanks to Duncan Haldane for his knowledgeable mentorship and Eric Wang for his bright and capable ongoing menteeship.

Thank you also to my lab mates for their assistance, advice, conversation, and most of all company, which made my graduate school experience so enjoyable. In particular, I would like to thank Liyu Wang, Austin Buchan, Ethan Schaler, Jessica Lee, Carlos Casarez, Anusha Nagabandi, Vanya Davydychev, Sareum Kim, Conrad Holda, and David McPherson.

Thank you to my friends and family for supporting me in all aspects of my life. Thank you to my parents, Mark and Laura Yim, and my brother, Kevin Yim, for their endless support, care, and love. Finally, a special thanks to Sarah Sterman.

Support was provided by the United States Army Research Laboratory under the Micro Autonomous Science and Technology Collaborative Technology Alliance and the Army Research Office Grant No. W911NF-18-1-0038.

Chapter 1

Introduction

1.1 Jumping motion

Legged robots offer unique capabilities and challenges when compared to other mobile platforms. Compared to wheeled or tracked platforms, legged platforms can move over extremely rough or discontinuous surfaces by taking advantage of isolated footholds. This can be achieved through a variety of gaits including walking, climbing, jumping, or hybrid motions incorporating aspects from several of these. Unlike heavier-than-air unmanned aerial vehicles (UAVs) that must produce lift, legged platforms are not subject to ground effect and do not produce downwash that might kick up dust or blow away light objects. Legged robots may be better suited than UAVs to operation involving intimate or forceful interaction with complex terrain. With different capabilities than either aerial vehicles or other terrestrial vehicles, improved legged robots could operate in environments that are currently very challenging, such as cluttered living spaces, buildings collapsed by disasters, cliff faces, or tree canopies.

Smaller robots can be cheaper and less intrusive, less restricted by small apertures, and more robust to falls and collisions [39]. However, with smaller locomotory appendages and a more limited reach, small terrestrial robots must use different strategies to move over their environment. Jumping is one such strategy that enables a locomotor to overcome obstacles or gaps longer than their maximum dimensions. This ability is particularly useful for small robots maneuvering in complex large-scale artificial or natural environments.

Ballistic motion planning can find jump sequences to navigate complex, discontinuous, high-slope environments [14]. Particularly aggressive trajectories include a chimney ascent, in which repeated bounces off of opposing vertical walls enable a jumper to climb up a narrow slot. Though these plans show what jumping could achieve, no currently existing robot platform is capable of executing the most aggressive jump plans, including chimney ascents in Earth gravity. Similarly, animals display acrobatic leaps unmatched by robots and provide natural examples of how effective high-performance jumpers can be.

Investigation of how insights from exemplary biological systems can inform robot design is one aspect of the rich field of biologically inspired robotics [53, 38]. Exemplary demonstrations of jumping 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 (Fig. 1.1). Robots capable of similar high performance leaping and landing could better approach the mobility of animals.

Jumping is a challenging mode of locomotion in which a locomotor transitions between free-fall ballistic flight phases and high-acceleration stance phases. During stance, a locomotor must balance on its feet while quickly accelerating its body; during flight, it must accurately orient itself for its next touchdown. Both estimation and control experience tight requirements on accuracy to achieve successful jumping motion in complicated environments.

Figure 1.1: Exemplary jumping animals. (Left) squirrel (image by Ed Sweeny via flickr, CC BY 2.0), (Center) goat (image via Good Free Photos, public domain), (Right) Human gymnast (image via Wikimedia Commons, public domain).

1.2 Contributions

This dissertation develops control and estimation algorithms for monopedal hopping. Salto-1P serves as the experimental platform to evaluate these algorithms. Earlier work using offboard motion capture for estimation and a classic Raibert hopper controller serves as the jumping-off point from which this work advances. Each chapter presents the development of a new ability for high-power monopedal hopping in chronological order: precise maneuvering over discontinuous surfaces in motion capture, operation without motion capture, and finally high-precision launching and landing transitions in chapters 2, 3, and 4 respectively.

Chapter 2 develops Deadbeat Foot Placement Hopping Control (DFPHC) to accurately land footfalls on desired footholds in a single jump [81]. By reducing foot placement error by more than a factor of three over an optimally tuned Raibert hopper controller, this control enables navigation around ledges and obstacles without missing a step and crashing or falling. Furthermore, numerical simulations show an approximately linearly relationship between jump vertical velocity and horizontal velocity sensitivity to touchdown angle errors, elucidating the relationship between jump height and foot placement accuracy.

Chapter 3 develops Spring Loaded Inverted Pendulum (SLIP) Hopping Orientation and Velocity Estimation (SHOVE) to enable fully autonomous hopping using only onboard sensing and processing [82]. Salto-1P demonstrated the first ever fully autonomous stable operation of an untethered, statically unstable monoped. In addition, Salto-1P demonstrated over 200 s of dead reckoning, attitude estimate disturbance rejection, operation on a compliant surface, and human joystick guidance using velocity commands.

Chapter 4 develops stance-phase balance control and a flight-phase touchdown plan to execute accurate leaps and balanced landings [83]. Using this control method, Salto-1P placed jumps longer than its bodylength with a 1.6 cm standard deviation, outperforming chapter 2, enabling jumps onto very narrow targets. Balanced landings enable transitions from jumping or hopping motion to a balanced standing posture on a narrow support. Finally, using reduced-order models, this work presents approximate bounds on acceptable horizontal velocity estimate and touchdown angle errors beyond which balanced landings are

unachievable.

1.3 Background

Foundations: Spring-loaded Inverted Pendulum and Marc Raibert

Significant work has addressed modeling and control of legged motion over the past four decades. The Spring-loaded Inverted Pendulum (SLIP) model developed in the 1980s serves as a useful reduced-order model that describes the motion of a locomotor's center of gravity (CG) as it runs or hops [12, 49].

In this model, a locomotor is considered to be a point mass attached to a massless linear spring that serves as a leg. While the spring is in contact with the ground, the locomotor is in stance phase and its CG is subject to forces from gravity and the leg spring. Usually, the end of the spring in contact with the ground is assumed to remain fixed by friction, which requires that the leg spring force remain inside the friction cone of the ground surface. Liftoff is the transition that marks the beginning of the flight phase when the spring force crosses zero and the leg loses contact with the ground. While in flight phase, the locomotor's CG experiences only the force of gravity and its motion is ballistic. During this time, the massless leg spring is assumed to be instantaneously re-orientable to any angle in preparation for the touchdown transition when the leg next makes contact with the ground.

This model captures the elastic energy storage and return often observed in running animals and replicated in many running robots. Furthermore, it captures how important the leg angle at touchdown is to the running gait. By appropriate selection of touchdown leg angle, the locomotor can hop at various horizontal speeds and vertical heights.

In 1984, Marc Raibert reported his foundational dynamically balanced monopedal hopping robot [58]. His control scheme decomposed the problem of balancing and directing a hopping monoped into three sub-tasks. The robot controlled its hopping height by changing its leg force during stance phase, its body angle by applying a hip torque in stance phase, and like the SLIP model, its horizontal velocity by setting the angle of its leg at touchdown. The work presented here draws heavily on both the SLIP-model and strategies of flight-phase leg angle selection for estimation and control.

Monopedal Hopping Robots

Many monopedal hopping robots have followed in the footsteps of Marc Raibert's robots. A single leg both enables hopping with very few appendages but also necessitates interesting locomotion like hopping or sliding. On the one hand, greater simplicity in mechanical construction and theoretical analysis may be possible due to the small number of appendages and contacts with the environment. On the other hand, without additional appendages, a single-legged robot may have difficulty executing motions other than hopping, such as walking, getting up, or even standing still in some cases. Prior jumping robots have addressed

different aspects of hopping motion using different strategies. This subsection discusses monopedal robots that generate large, repeated hops. A comprehensive overview of earlier monopedal robots is provided by [64]. The discussion here does not include hoppers with large feet that make small hops, robots that execute one very large jump before resting on the ground, or robots confined to a one-dimensional vertical rail.

Many monopeds are constrained to an approximate vertical plane to simplify their dynamics. Uniroo by Garth Zeglin was an early monoped with kinematics inspired by a kangaroo [85]. The later Kenken was inspired by a dog [37]. The ARL Monopod [28] and ARL Monopod II [4] demonstrated high running speed and low specific resistance in the late 1990s. More recent works have used boom-mounted monopeds for experimental validation of SLIP approximations and controllers [75, 45, 73]. The Penn Jerboa has two legs, but is based very intentionally on the SLIP model composed with an inertial tail, which it uses to energize its jumps [18].

Often boom-mounted monopeds serve as an intermediate step in the development of a multi-limbed robot. Thumper [56] is a monopedal version of the boom-mounted biped MABEL [30] and ATRIAS 1.0 [29] is a monopedal version of the 3D-capable biped ATRIAS [36]. Both of these series of robots intentionally incorporate SLIP-like properties into their design.

Other free-hopping monopeds have served as interesting and challenging experimental platforms. Garth Zeglin's subsequent bow-leg hopping robot [84, 86] uses a bow-spring leg and inspired following platforms like the air table wall-jumping parkourbot [19]. Linear Elastic Actuator in Parallel (LEAP) used onboard sensors to hop and balance, but was not stable indefinitely [9].

1.4 Salto and Salto-1P

Salto and its successor Salto-1P are small untethered monopedal hopping robots developed in the Biomimetic Millisystems Lab at the University of California, Berkeley (Fig. 1.2 left and right respectively). Development of these robot platforms paved the way for the work of this dissertation. This subsection details work led by the author's collaborators Duncan Haldane and Mark Plecnik in [34] and [55].

Salto

Galago senegalensis

The construction of Salto was inspired by the extraordinary jumping ability of the lesser bushbaby, *Galago senegalensis* (Fig. 1.3 left) as analyzed by Aerts [2]. This small primate is only about a third of a meter from snout to extended toe, but jumps two meters high. To make these jumps, the legs of *Galago* use a strategy of power modulation, called "power amplification" in biological literature. Power modulating systems use elastic elements in combination with an actuator (muscle) to deliver peak power higher than the maximum

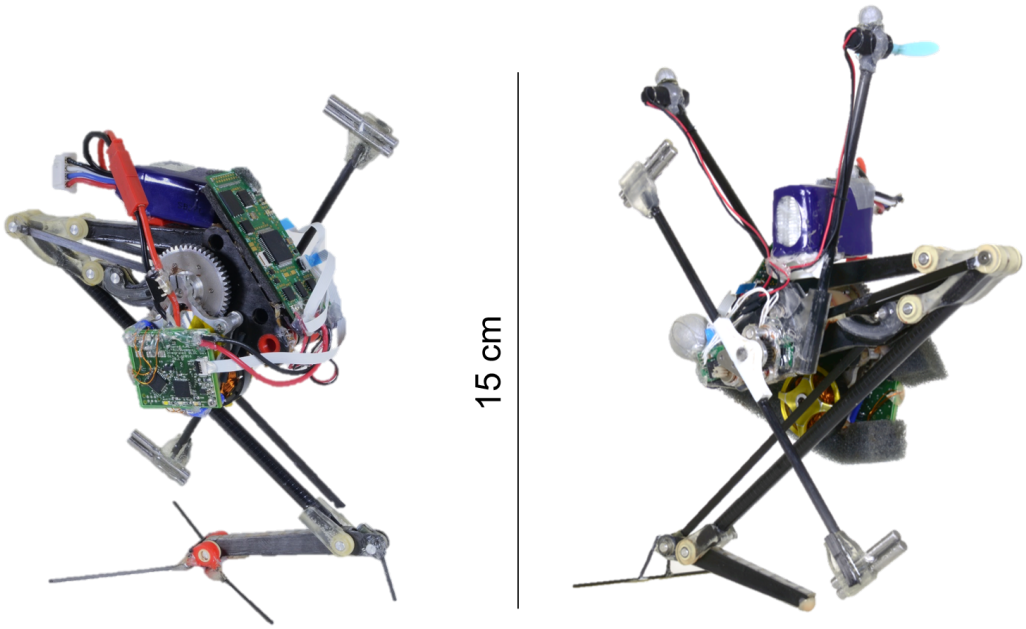


Figure 1.2: Salto and Salto-1P. Right image by Ethan Schaler.

output of the actuator alone. *Galago* powers its jumps mostly with the vastus muscle of the leg, attached across the knee by an elastic tendon. At the beginning of a jump, action of the vastus mostly serves to stretch the series-elastic tendon, storing elastic potential energy. As the jump advances, the tendon contracts in concert with the muscle to deliver fifteen times more power than the muscle could alone, accelerating *Galago* very quickly (Fig. 1.3 right). Just as an aircraft must accelerate faster to reach liftoff speed on a shorter runway, so must a smaller jumper accelerate faster to jump from its shorter legs. High output power is a must for small jumpers whose stance durations are limited by the length of their comparatively shorter legs.

However, high power and its associated high accelerations are useful for more than making a compact jumper. Dynamic maneuvers like jumping off of steep slopes and vertical walls must be executed quickly if a jumper is not to slide or tumble (unless it is attached to the wall somehow). Greater power and capacity to accelerate give jumpers freedom to quickly accelerate off of steeper slopes before they tumble from their footholds under the pull of gravity. Furthermore, rapid motions can be chained together into other rapid maneuvers like running and bouncing.

Vertical Jumping Agility and Series-Elastic Power Modulation

To compare rapid jumping by animals and robots, Duncan Haldane proposed the Vertical Jumping Agility metric [34]. This metric is defined as the ratio of a jumper's vertical CG

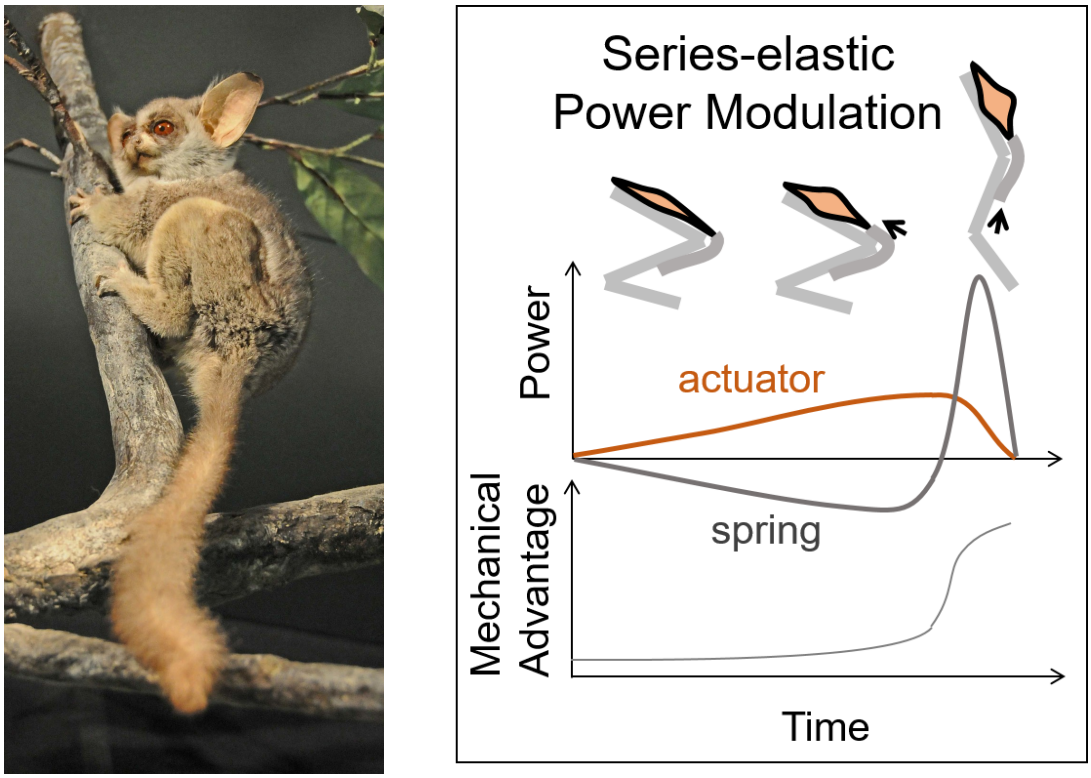


Figure 1.3: (Left) the lesser bushbaby, *Galago senegalensis* (image from Smithsonian Open Access, public domain), (Right) illustration of the principle of Series-elastic Power Modulation.

displacement from initial pose to flight apex divided by the time to apex starting from the initial application of actuator power. Vertical Jumping Agility is measured in vertical meters per second and can be interpreted as the rate at which a jumper could climb a ladder with rungs placed at its jump apexes. Many robots produce either extremely high, or very fast jumps, but none can match the total vertical jumping agility of *Galago*.

To build a robot with a vertical jumping agility approaching that of *Galago*, Duncan Haldane developed the Series Elastic Power Modulation (SE+MA) mechanism design paradigm. Here, a series-elastic actuator drives a mechanism with a variable mechanical advantage in order to store and return elastic potential energy and boost the mechanism peak power output. In the initial configuration, the mechanical advantage between the series-elastic actuator and the mechanism output is low, meaning that large torques at the actuator produce small forces at the robot's foot. This is where the initial power storage phase of the jump occurs. Due to the low mechanical advantage, actuator action mostly stretches the spring, storing elastic energy, while the mechanism very slowly extends. As the mechanism extends and its configuration changes, the mechanical advantage rises dramatically and the jump transitions into the power return phase. As in *Galago*'s jump, the actuator and series-elastic

spring together provide a higher peak power than the actuator could have provided alone.

Design of Salto

For actuation, Duncan Haldane built a compact and lightweight series-elastic actuator assembly consisting of a Scorpion S-1804-1650KV brushless motor with custom winding for a higher motor torque constant, a 25:1 aluminum gearbox, and a conical-tapered latex spring with geometry adapted from [61]. A custom motor driver board commutates the motor and a small 3S (three cells in series) Lithium polymer battery provides power.

Mark Plecnik led the development of the variable mechanical advantage leg mechanism. The result is a single degree of freedom planar eight-bar linkage developed using design exploration and kinematic tuning described in [55]. This leg mechanism achieves several objectives. It must exhibit the low mechanical advantage when contracted and high mechanical advantage when extended for SE+MA. Furthermore, it must be physically buildable without intersecting the ground beneath the robot's foot or generating excessive impulses or internal loads. In addition, it is beneficial to have a long output leg extension stroke (to increase the duration of the stance phase) and a large input crank stroke (to decrease the torque required of the motor). Finally, similarity to the SLIP model is achieved by constraining the foot to move along a straight line of action and by mass balancing the mechanism so that extension and retraction of the leg generates no reaction moments on the body. Duncan Haldane translated Mark Plecnik's leg kinematics design into a physical prototype built primarily out of carbon fiber composite honeycomb, fiberglass, unidirectional pultruded carbon fiber, and polyurethane cast components, assembled with aluminum pins riding in polymer bushings at revolute joints.

The final integrated result is the robot Salto. Salto demonstrated the highest vertical jumping agility among untethered electric robots at the time of publication at 1.75 m/s, 78% of *Galago*. With its high vertical jumping agility, it made the first robotic demonstration of a wall jump in Earth gravity. In this maneuver, Salto leapt off of the ground towards a vertical wall, then made a second quick leap off of the wall to gain additional height. By making the wall jump, Salto propelled itself an average of 20% higher than would have been possible using a single jump from the ground alone.

Salto-1P

To investigate continuous operation rather than one or two jumps, Salto needed a way to control its leg angle in all directions. Salto is equipped with a balanced inertial tail (flywheel) to control its angle in one direction, but without a way to control or at least stabilize its angle about any other axis, it inevitably falls over. This led to the development of Salto-1P with actuation and construction led by Duncan Haldane and hopping control by the author [32].

Salto-1P builds on the same leg mechanism at the heart of Salto. Its reworked construction is lighter than its predecessor's and it has a new small diameter rubber toe instead of

the cross-shaped foot that Salto used for sitting in its starting crouch. Increased power is provided to the motor by a higher voltage custom 4S Lithium polymer battery. Salto-1P achieves a higher vertical jumping agility, 1.83 m/s, than Salto. Most importantly, Salto-1P is equipped with two small aerodynamic thrusters: miniature quadrotor propellers each driven by the same type of 7mm diameter brushed motor that drives the tail.

Salto-1P reference frames

Salto-1P's attitude is parameterized using ZXY Euler angles called yaw, roll, and pitch respectively. The leg mechanism lies in the X-Z plane with its line of action along the body-fixed Z-axis, coincident with the robot's CG (Fig. 1.4 left). The axis of the balanced inertial tail is parallel to the body-fixed Y-axis, so that torques applied to the tail accelerate the body about this axis. The two thrusters both point in the positive Y-direction and are displaced 8 cm above the CG along the Z-axis. One thruster is displaced 4 cm forward of the CG along the X-axis and the other 4 cm behind. Each thruster can apply between -5 and 7 grams force in the +Y direction (forces are asymmetric since the quadrotor propellers have non-zero camber). The sum of the thruster's forces applies a roll torque, while the difference of their forces applies a yaw torque. Together with the tail, they provide full actuation of the robot's attitude.

Salto-1P electronics, actuation, and sensing

Like Salto, Salto-1P is controlled by the ImageProc robot control board which carries a dsPIC33FJ128MC706A microcontroller, a MPU6000 Inertial Measurement Unit (IMU) with triaxial rate gyroscopes and triaxial accelerometers, flash memory, power management, four H-bridge outputs, and an XBee Radio. Salto-1P's leg motor and three attitude actuators make a total of four actuation control signals. Besides its IMU, Salto-1P uses encoders to measure its leg linkage configuration, leg motor angle, and tail angle. From its linkage configuration, Salto-1P can measure its leg length and by combining its linkage configuration with its leg motor angle, it can measure its series-elastic spring deflection. Spring deflection is related to force along the leg line of action; a significant positive force implies that the robot is in contact with the ground, serving as a touchdown and liftoff sense.

Salto-1P performance

Now capable of fully controlling its orientation at touchdown, Salto-1P ran under a Raibert-like hopping controller to explore its repetitive hopping performance. As intended, Salto-1P exhibits an extremely high acceleration during stance phase and a very short stance time (Fig. 1.4 right). Stance accelerations average nearly 14 times earth gravity; as a consequence, stance lasts only about 70 ms. Combined with Salto-1P's high jumps and long flight times, it can spend a mere 7.7% of its stride in contact with the ground, a ratio termed the duty factor in biological literature.

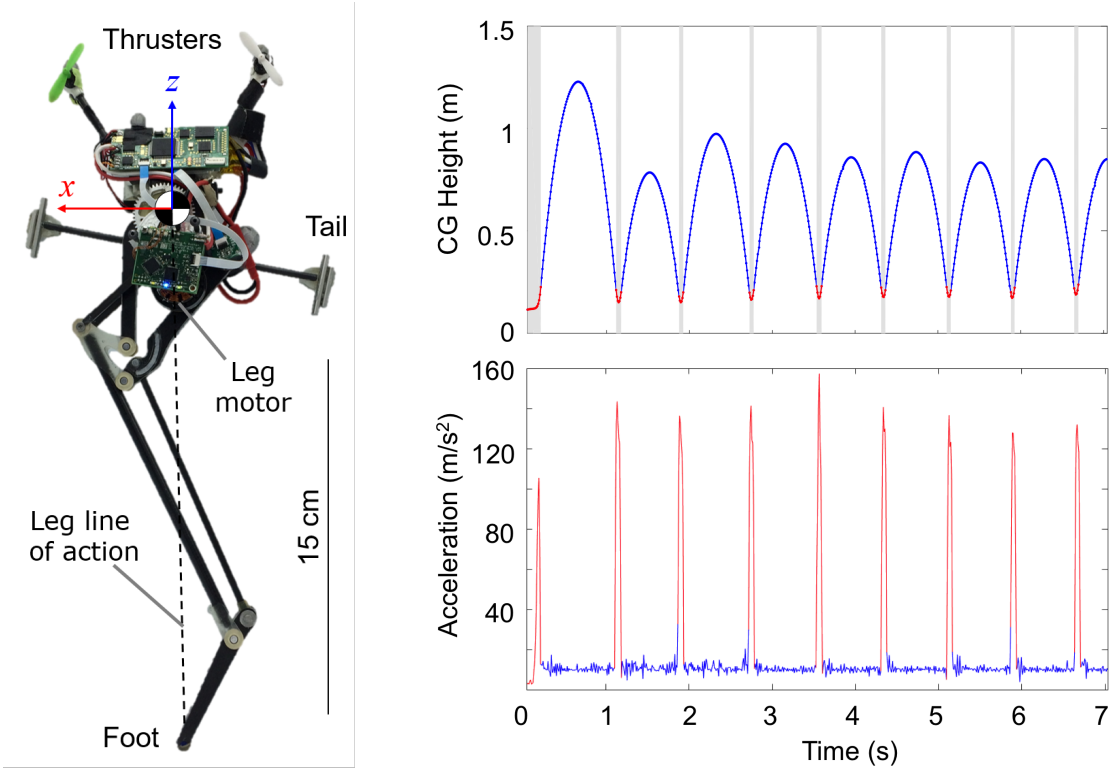


Figure 1.4: Salto-1P and its hopping performance

This high acceleration and short stance time is important for two reasons. First, the short stance phase makes a monoped approximately as well suited to hopping motion as a multi-legged locomotor. While multiple legs allow the exertion of torques during stance phase, the stance phase duration is so short and its acceleration so high, that this ability would be difficult to use. Note that a lack of additional appendages does complicate landings in cases where a multi-legged locomotor could splay its feet for stability, as addressed in chapter 4. Second, it is this high acceleration and short stance time that first enabled Salto to perform its wall jump. With a stance acceleration of 14 gs, the acceleration due to gravity can nearly be neglected and the orientation of the ground's normal vector relative to gravity is less important. With a sufficiently high acceleration capacity, a jumper can leap off of a vertical or even inverted surface just as well as off of a horizontal one.

With a highly capable experimental platform developed and tested, this dissertation presents the development of estimation and control to best use high jumping performance to move over discontinuous surfaces in and out of the lab.

Chapter 2

Precise Hopping

2.1 Introduction

Campana and Laumond developed a ballistic motion planning algorithm for saltatorial locomotion that respects takeoff velocity and friction cone limits in complicated terrain [14]. The planner is able to find trajectories across disparate stepping stones or even up channels between vertical walls. Using trajectories like these, jumping robots could traverse terrain that could be extremely difficult for other kinds of platforms. However, following these trajectories in complex terrain requires very accurate foot placement to avoid missing a foothold. Furthermore, a robot must redirect its ballistic trajectory towards the next foothold using only one stance phase. This single-stance control is akin to deadbeat control of a discrete-time system and I will likewise call a controller with this ability a Deadbeat Foot Placement Hopping Controller (DFPHC). This work aims to improve the foot placement precision of a small jumping robot by developing deadbeat foot placement hopping control and examines the effect of attitude error on foot placement precision. [1]

The Spring Loaded Inverted Pendulum (SLIP) [12] is a common model for running and hopping; however, the SLIP dynamics cannot be integrated to produce a closed-form solution except for particular non-linear spring forces without gravity. In [62] a closed-form integrable SLIP-like model is used to derive deadbeat control of hopping apex state. Many approximations to the SLIP dynamics like [65], [27], and [63] have also been proposed to sidestep the non-integrability.

Numerical simulation of the hopper dynamics is another approach. In [59], a lookup table and a polynomial approximation to the table selected control inputs. In [15], [78], and [67] deadbeat hopping of SLIP-like models is examined analytically and in simulation. In addition, sensitivity of a deadbeat controller to model error and control input error was examined in [78], finding that more aggressive turns produced larger errors. Without using SLIP dynamics, [5] and [23] investigated sequencing deadbeat control of monopedal jumping platforms. Clocked deabeat control of vertical hopping without horizontal dynamics was derived in [16]. A reachability approach controlled the next apex state during the stance phase in [54].

Several works have demonstrated precise foot placement in physical experiments. Hodgins demonstrated precise foot placement to enable stair climbing with a boom-mounted planar robot by varying the horizontal velocity, flight time, and stance time to control step length[35]. In [73] partial feedback linearization demonstrated high precision foot placement also on a boom-mounted robot.

[1] This chapter is revised from work presented as "Precision jumping limits from flight-phase control in Salto-1P," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018 [81]. This material is based upon work supported by the United States Army Research Laboratory under the Micro Autonomous Science and Technology Collaborative Technology Alliance and the Army Research Office Grant No. W911NF-18-1-0038.

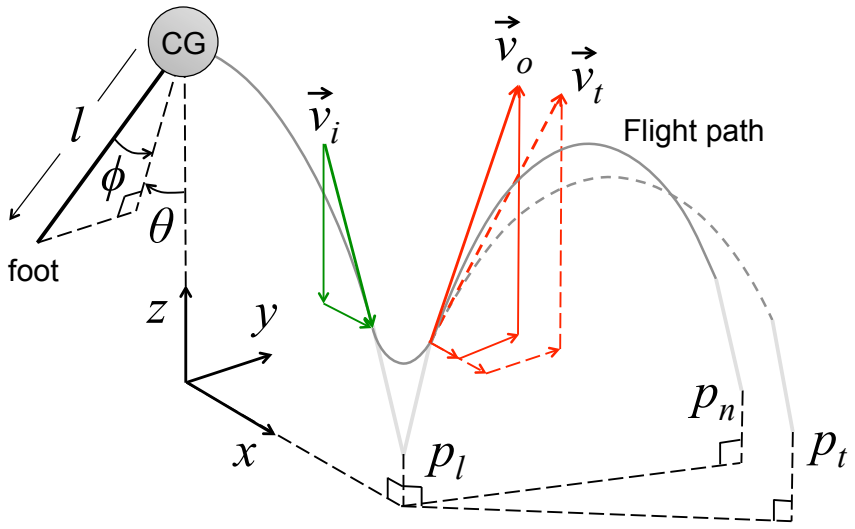


Figure 2.1: Reference frames and variables.

2.2 Methods

Approach

In developing a deadbeat foot placement hopping controller, I consider a SLIP-like hopping model shown in Fig. 2.1. Jumping motion can be divided into alternating stance phases (stances) and flight phases (flights). Touchdown is the transition between flight and stance when the foot strikes the ground at a foot placement point. Takeoff is the transition between stance and flight when the foot leaves the ground.

During flight, a jumping robot has no control over the motion of its center of gravity (CG) without specialized means to apply large forces in the air. Its CG trajectory in flight (flight path) is a ballistic parabola set by its previous foot placement and velocity at takeoff. The robot has only a little control over its next foot placement and velocity at touchdown through reorienting its foot relative to its CG.

To reach a desired foothold, the robot must set its velocity at takeoff to aim its flight path towards the foothold. Takeoff velocity can be changed either by control action during stance or by changing stance initial conditions like leg angles at the previous flight's touchdown. [32] demonstrated that this latter technique was useful for robots like Salto-1P that have stances too fast for effective closed-loop control. Thus to achieve deadbeat foot placement hopping control, a robot can set touchdown conditions from its previous flight to aim its next flight path towards the desired foothold.

I split the deadbeat foot placement hopping control into two parts. The first part, a velocity planner, determines what takeoff velocity $\vec{v}_t$ will take the robot from its upcoming foot placement p_l and place its next foot placement p_n on the desired foothold at p_t. Since

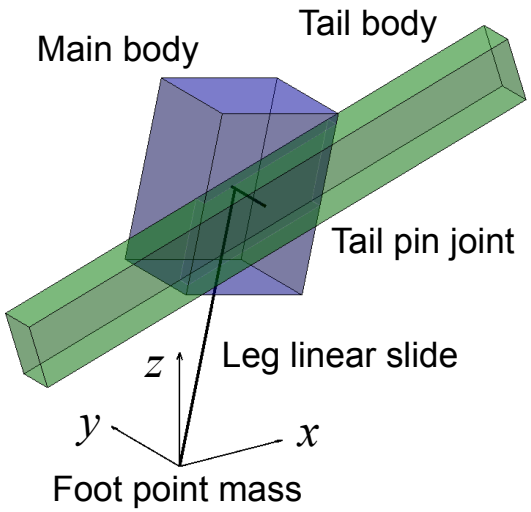


Figure 2.2: Simulation model.

a family of ballistic parabolas pass through two points, the velocity planner has one free variable that it can set by selecting a parameter like vertical velocity at takeoff or apex height.

The second part, a velocity controller, determines appropriate touchdown conditions so that the stance phase will start from the touchdown velocity $\vec{v}_i$ and end at the takeoff velocity $\vec{v}_o$ so that $\vec{v}_o = \vec{v}_t$ computed by the velocity planner. Salto-1P's touchdown is parameterized by the roll angle (ϕ) pitch angle (θ), and leg length (l). Touchdown leg length changes the impulse delivered on the ground: a shorter initial length provides a larger net impulse. I parameterize the robot's attitude with ZXY Euler angles, but ignore the yaw angle (ψ) since the SLIP-like model is symmetric about Z.

I assume the robot's leg length is small relative to its jump height as is the case for Salto-1P. Therefore, the effects of ϕ, θ, and l on p_l and $\vec{v}_i$ can be ignored so that the velocity controller's actions do not affect the velocity planner.

Simulation Model

In order to predict $\vec{v}_o$ resulting from certain touchdown conditions, I simulated stance in a custom Matlab rigid body simulation matched to the physical parameters of Salto-1P. This model consists of three bodies shown in Fig. 2.2: a rigid body for the main body of the robot, a rigid body for the tail connected to the main body by a pin joint, and a point mass on a linear sliding constraint relative to the main body to serve as the foot and approximate the mass and inertia of the leg.

Leg force is computed with a leg friction model and nonlinear spring model from [33],

the kinematic relationship between the crank angle and the foot extension produced by the 8-bar linkage, and a DC motor model.

The touchdown transition is modeled as an inelastic collision between the point mass foot and the rigid ground. Takeoff is another inelastic collision in which the stance leg length reaches its maximum extension and momentum transfers from the main body to the foot.

Salto-1P's moment of inertia about its lateral and longitudinal axes are both approximately 130×10^{-6} kg m^2. Since the robot weighs 0.103 kg and its CG is 0.10 m above its foot with the leg fully retracted, the robot's moment of inertia about its foot is dominated by the CG distance from the foot and is not significantly changed by the robot body's heading. Since the robot's foot moves along a straight line coincident with the CG and its moment of inertia is nearly the same at all headings, the robot's stance phase is insensitive to heading and its touchdown yaw angle can be neglected.

Furthermore, since the balanced inertial tail's angular velocity is kept low by braking during stance phase, the tail's angular momentum is small compared to the angular momentum due to the motion of the robot's CG. As with the robot's yaw heading, the tail angle and angular velocity are also neglected. With the above assumptions, the robot's behavior is similar to a SLIP-like point mass with a motor-controlled leg force and I can reduce the number of parameters that must be searched in simulation.

Adopting the convention of [78], $\vec{v}_i$ is measured aligned with the horizontal bearing of the robot's velocity. In this touchdown velocity-aligned frame, $\vec{v}_i$ has only two elements: longitudinal (v_{ix}) and vertical (v_{iz}). $\vec{v}_o$ has three elements: longitudinal (v_{ox}), lateral (v_{oy}), and vertical (v_{oz}). This simulation approximates a map from $\vec{v}_i$ to $\vec{v}_o$ controlled by ϕ, θ, and l, $\vec{v}_o = f_s(\vec{v}_i, \phi, \theta, l)$.

Velocity Controller

The velocity controller is composed of three functions whose inputs are the touchdown velocity and desired takeoff velocity $\phi = f_\phi(\vec{v}_i, \vec{v}_t)$, $\theta = f_\theta(\vec{v}_i, \vec{v}_t)$, $l = f_l(\vec{v}_i, \vec{v}_t)$.

From the SLIP-like model's rotational symmetry about the Z-axis and the coordinate frame selection, there are several symmetries in the velocity control functions:

- f_θ is odd in x ($f_\theta(v_{ix}, v_{iz}, v_{tx}, v_{ty}, v_{tz}) = -f_\theta(-v_{ix}, v_{iz}, -v_{tx}, v_{ty}, v_{tz})$) and even in v_{ty}.
- f_ϕ is odd in v_{ty} and even in x.
- f_l is even in x and v_{ty}.

I ran simulations of stance for a grid of 16,170 touchdown conditions. The baseline θ resolution was 0.05 rads and the ϕ resolution was 0.03 rads but these became finer and confined to a smaller deviation for larger v_{iz} since large deflections would have caused slipping. The horizontal and vertical velocities were swept at a constant step of 0.4 m/s from -2.2 to 2.2 m/s and -2.0 to -4.4 m/s respectively. l varied from 0.226 m to 0.25 m at a constant step of 0.004 m. Simulations for which the main body impacted the ground, reached velocities

outside the range of the touchdown velocities, and those for which the foot would slip on a flat horizontal surface with coefficient of friction 1 were discarded.

Similarly to [59], I use a polynomial approximation for the velocity controller functions. I fit five-dimensional third-order Taylor series approximations for f_θ, f_ϕ, and f_l centered at the point $v_{ix} = v_{ox} = 0$ m/s, $-v_{iz} = v_{oz} = 3.3$ m/s and $v_{oy} = 0$ m/s using the simulation data. This point corresponds to hopping in place with an apex height of 0.55 m. Due to the symmetry constraints outlined above, there are 31 non-zero coefficients in total: five linear, eleven quadratic, and fifteen cubic terms. The final velocity controller f_θ function is given below as an example:

$$\begin{aligned} f_\theta(v_{ix}, v_{iz}, v_{tx}, v_{ty}, v_{tz}) = &-0.2071 v_{ix} + 0.078 v_{tx} \\ &- 0.0385 v_{ix} v_{iz} + 0.0196 v_{tx} v_{iz} \\ &- 0.0052 v_{ix} v_{tz} - 0.04 v_{tx} v_{tz} \\ &+ 0.003 v_{ix}^3 - 0.036 v_{tx}^3 \\ &- 0.0024 v_{ix} v_{ty}^2 - 0.0046 v_{tx} v_{ty}^2 \end{aligned} \tag{2.1}$$

After the deadbeat velocity controller computes the desired θ and ϕ angles, they are rotated out of the touchdown velocity-aligned frame to the ZXY Euler angle frame.

Velocity Planner

The flight estimation and velocity planner determine the desired takeoff velocity v_t at the end of the upcoming stance so that the robot next lands at p_t. For convenience, the height of p_l is assumed to be the height of the previous desired foot placement. p_l is then predicted as the point at which the flight path reaches this height.

The velocity planner computes the desired flight time t_f from v_{tz} and the heights of the foot placements p_{tz}, and p_{lz}.

$$t_f = \frac{v_{tz}}{g} + \frac{\sqrt{v_{tz}^2 - 2g(p_{tz} - p_{lz})}}{g}$$

v_{tx} and v_{ty} are then the horizontal displacement of p_t from p_l divided by the desired flight time. These velocities are rotated into the touchdown velocity-aligned frame.

$$v_{tx} = \frac{p_{tx} - p_{lx}}{t_f}, \; v_{ty} = \frac{p_{ty} - p_{ly}}{t_f}$$

Note that I ignore the foot deflection away from the CG trajectory as stated earlier. In the results I show that this error is small in comparison to other errors.

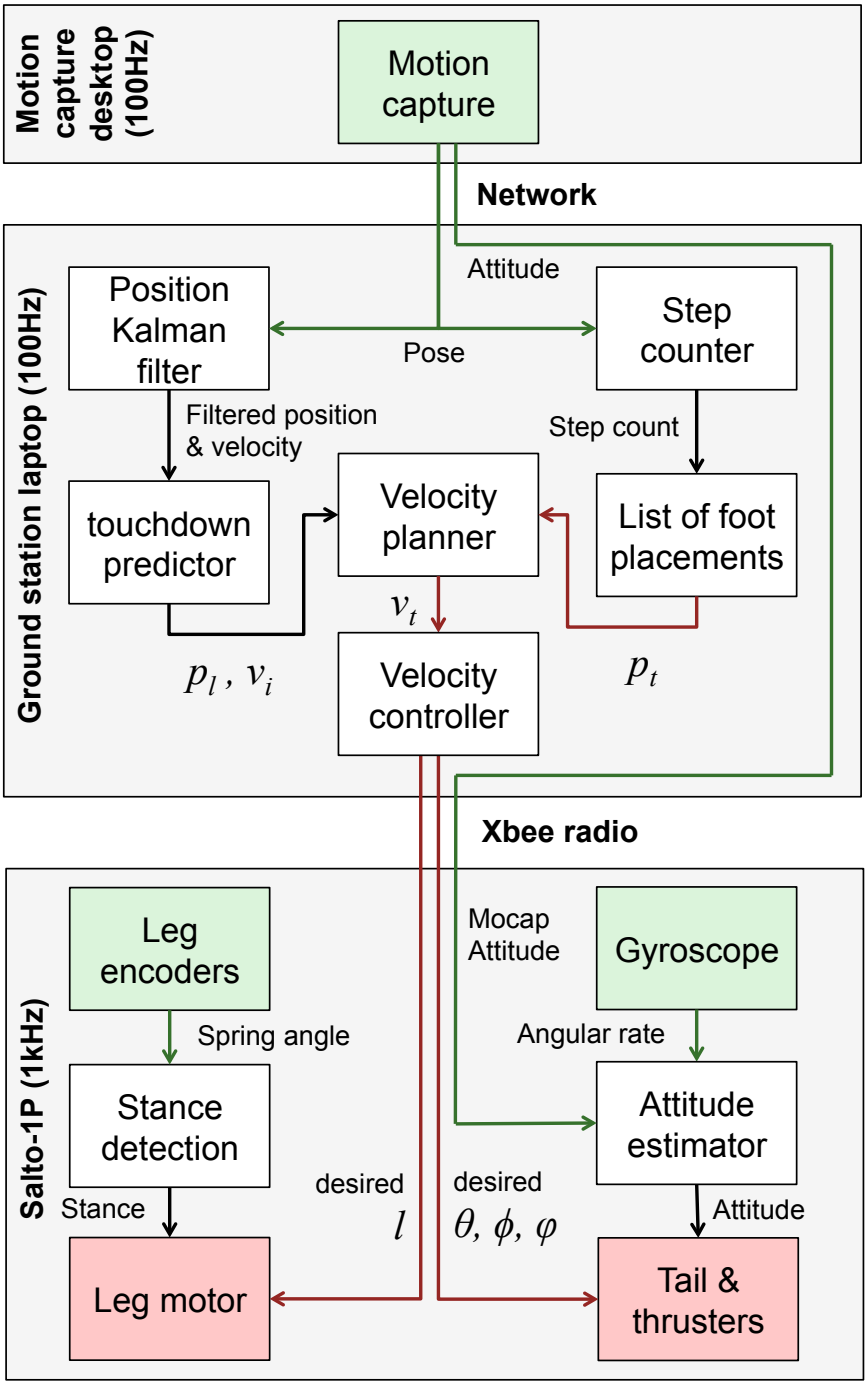


Figure 2.3: Block diagram of experimental system, deadbeat velocity planning, and deadbeat velocity control. The Raibert controller replaced the "Velocity controller" block during comparison experiments.

Physical experiments

The Salto-1P robot was used for the following experiments. It can jump 0.90 m high with a vertical jumping agility of 1.36 m/s [34]. The robot is 0.17 m tall with leg retracted and 0.32 m tall with leg fully extended.

Optitrack Prime 13 cameras and Motive software track retroreflective markers on the robot. As shown in Fig. 2.3, motion capture data stream at 100 Hz to a ground station laptop running Robot Operating System (ROS). The ground station uses a Kalman filter to estimate the robot's linear position and velocity in flight and uses these to predict p_l. Acceleration is estimated by double-differentiation of position. Touchdown is detected when acceleration exceeds a threshold. The Kalman filter is disabled in stance and reinitialized shortly after takeoff.

For each experiment, I specify a list of desired foot placements to define the desired hopping trajectory. Each desired foot placement has five parameters: x, y, and z position of p_t, desired yaw angle, and desired vertical velocity at takeoff.

The step count increments by one upon touchdown. The next desired foot placement in the list of desired foot placements is then selected as the new p_t. The velocity planner computes v_t from p_l and p_t. The velocity controller computes the desired ϕ, θ, and l.

The groundstation computer sends the attitude measured by the motion capture system, desired touchdown attitudes, and desired leg length to the robot via the XBee radio.

The robot's onboard microcontroller runs asynchronously from the ground station. Three 1 kHz proportional derivative (PD) controllers on yaw, roll, and pitch angles activate the thrusters and tail in flight to stabilize the robot's attitude to the desired touchdown leg angles. Another 1 kHz PD controller retracts the leg to the desired length. The microcontroller integrates rate gyroscope readings from the onboard IMU to estimate attitude. It fuses this estimate with the received motion capture attitude measurement to prevent drift due to accumulation of integrated gyroscope error.

The robot also detects its own touchdown by measuring the deflection of the series elastic spring. If this exceeds a threshold, the robot determines that it has hit the ground and it extends its leg. Takeoff is detected when the spring deflection reaches zero or the leg reaches its full extension.

After each experiment, I manually synchronize recorded groundstation and robot data by aligning motion capture measurements of robot height with onboard measurements of leg deflection during stance.

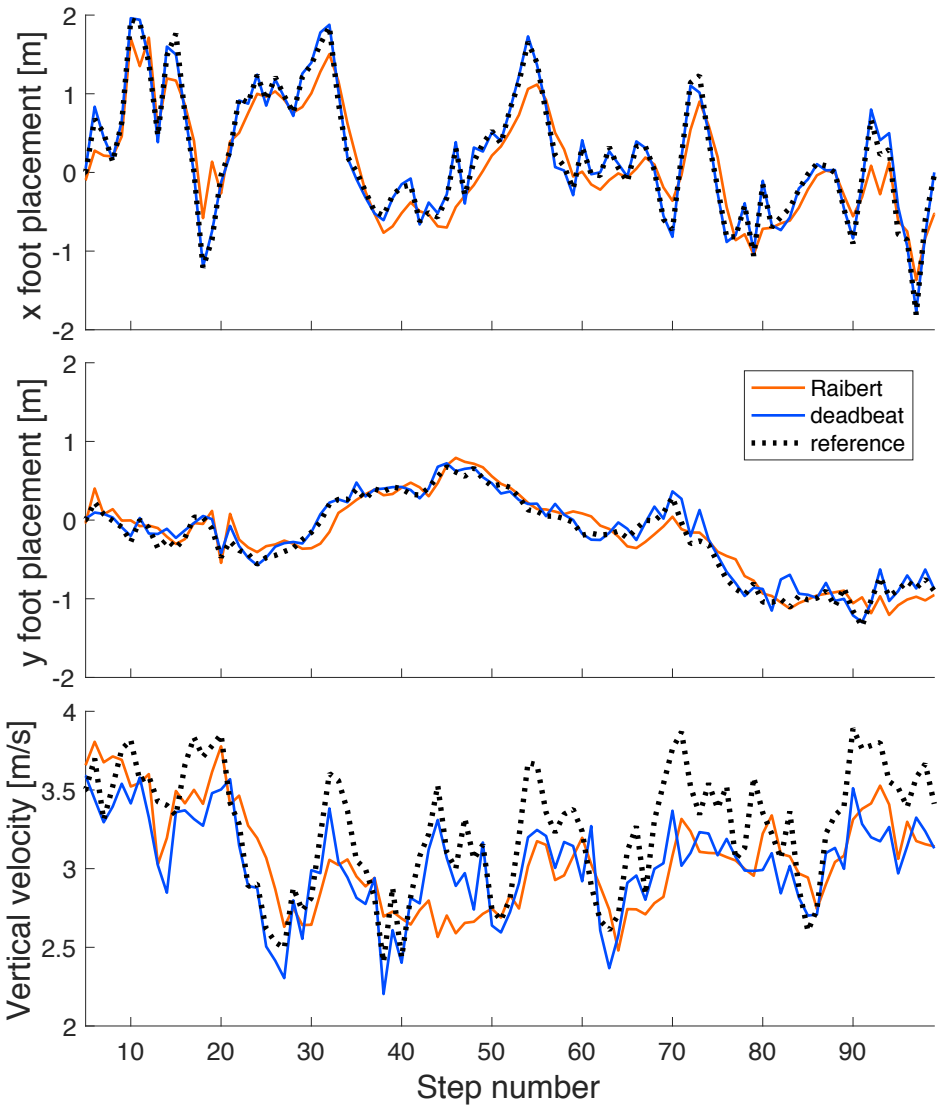


Figure 2.4: Experimental comparison of deadbeat velocity controller and Raibert hopper controller foot placement error.

2.3 Results

Foot Placement Accuracy

To evaluate foot placement accuracy, the robot was commanded to follow a series of 95 foot placements in a random walk on flat ground with a yaw heading of zero and random vertical velocity changes. The commanded and achieved foot placement trajectories are shown in

Longitudinal Error [m]	STD	95th percentile
Previous controller [32]	0.516	1.0
Optimal Raibert	0.306	0.6
Deadbeat	0.097	0.3

Table 2.1: Experimental longitudinal foot placement accuracies for different controllers. For comparison, data from [32] were collected from a slightly less agressive run with only longitudinal jumps commanded.

Fig. 2.4. Longitudinal (x) displacements ranged up to 1.3 m and lateral (y) displacements ranged up to 0.36 m corresponding to longitudinal speeds up to 1.68 m/s and lateral speeds up to 0.43 m/s. The longest jumps were over four times the robot's body length. The largest change in horizontal velocity in one stance was 2.93 m/s. The longitudinal jumps are larger than the lateral because Salto-1P's inertial tail has greater control authority than its lateral thrusters. Horizontal jump direction angle change covered all angles from -180 degrees to +180 degrees. Desired vertical velocity at takeoff changed by up to 0.5 m/s higher or lower on each jump.

This experiment compared the performance of the deadbeat velocity controller and an optimally tuned Raibert controller. This Raibert controller replaced the "Velocity controller" block in Fig. 2.3. Since the Raibert horizontal velocity controller is parameterized only by one feedback gain, this parameter was optimized by hand tuning. Thrust control was selected by looking up l based on v_{tz} in a linearly interpolated lookup table of six experimentally measured pairs of leg length and steady-state hopping vertical velocity. In the canonical Raibert hopper controller, the stance duration is assumed to be the same as the last stance duration, but I used a second linearly interpolated lookup table of steady state hopping vertical velocity and stance duration since measuring stance duration was difficult for the motion capture system.

Fig. 2.5 compares the foot placement error $p_n - p_t$ and vertical velocity error $v_{oz} - v_{tz}$ of the Raibert controller and deadbeat velocity controller. The Raibert controller achieved a foot touchdown error standard deviation of 0.306 m longitudinally and 0.156 m laterally, while the deadbeat velocity controller achieved a foot touchdown error standard deviation of 0.092 m longitudinally and 0.097 m laterally.

Both controllers experienced decreased hopping height over time as the robot's battery voltage dropped during the run. The Raibert hopper controller was less able to follow quick changes from one v_{tz} to the next and rounded off the corners of v_{oz}. In comparison, though its v_{oz} had a steady offset below v_{tz} as the battery depleted, the deadbeat velocity controller better adjusted v_{oz} in a single stance phase. The Raibert hopper controller achieved a vertical velocity error standard deviation of 0.29 m/s while the deadbeat velocity controller achieved a vertical velocity error standard deviation of 0.20 m/s.

Foot placement precision is important since this defines the foothold size that the robot

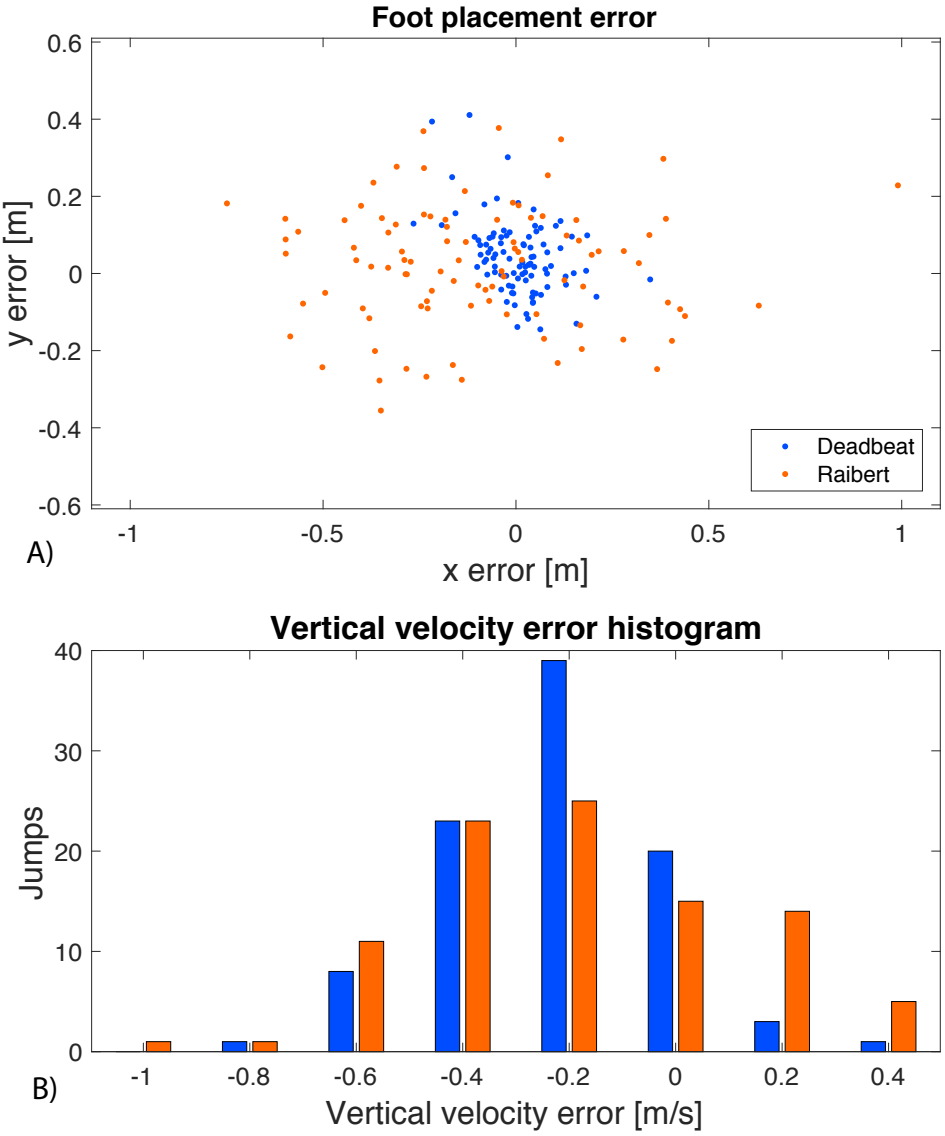


Figure 2.5: Experimental deadbeat velocity controller and Raibert hopper controller foot placement error and vertical velocity error.

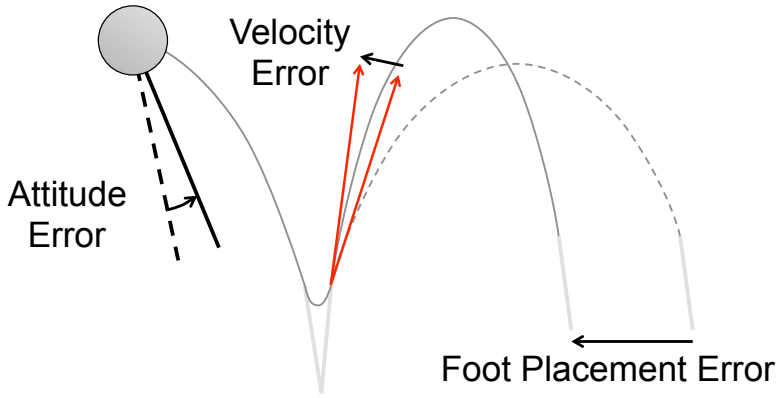


Figure 2.6: Propagation of touchdown leg angle error to foot placement error.

can reasonably hit. The Raibert controller placed 95 percent of its touchdowns within 0.6 m of the desired position, while the deadbeat velocity controller placed 95 percent of its touchdowns within 0.3 m of the desired position.

Foot Placement Sensitivity to Touchdown Leg Angle

Like [58], [66], [78], [67] and others, my algorithm controls hopping by selecting touchdown leg angles and leg lengths during flight. Consequently, the foot placement accuracy is sensitive to the flight attitude control accuracy used to position the leg as shown in Fig. 2.6. To investigate how ϕ and θ accuracy would affect foot placement accuracy, I examined the sensitivity of $\vec{v}_o$ to the touchdown conditions. Using the simulation results that generated the deadbeat velocity controller, I calculated sensitivity by discrete differentiation of $\vec{v}_o$ with respect to $\vec{v}_i$, ϕ, θ, and l. This approximated the Jacobian of the stance velocity map f_s at each point in the grid of simulated initial conditions.

As expected, takeoff longitudinal velocity v_{ox} is highly sensitive to θ, and takeoff lateral velocity v_{oy} is highly sensitive to ϕ. v_{ox} is insensitive to ϕ and v_{oy} is insensitive to θ. These sensitivities are the coefficients by which leg angle errors will propagate into the takeoff horizontal velocities and foot placements.

Importantly, the sensitivity of takeoff horizontal velocity to touchdown leg angles increases as the robot's touchdown vertical velocity v_{iz} and takeoff vertical velocity v_{oz} become faster. Fig. 2.7A shows this sensitivity plotted against v_{iz}. The relationship is approximately linear with some upwards curvature. In steady forward hopping at $v_{ix} = v_{ox} = 1.0$ m/s and $-v_{iz} = v_{oz} = 2.4$ m/s, v_{ox} varies at 0.13 m/s per degree of θ deflection and v_{oy} varies at -0.13 m/s per degree of ϕ deflection. For higher steady forward hopping at $-v_{ix} = v_{ox} = 1.0$ m/s and $v_{iz} = v_{oz} = 4.0$ m/s, the horizontal velocity sensitivity magnitudes are 0.35 m/s per degree. The magnitudes of the v_{ox} sensitivity to θ and the v_{oy} sensitivity to ϕ remain very close to each other throughout the operating domain, so only the longitudinal direction is

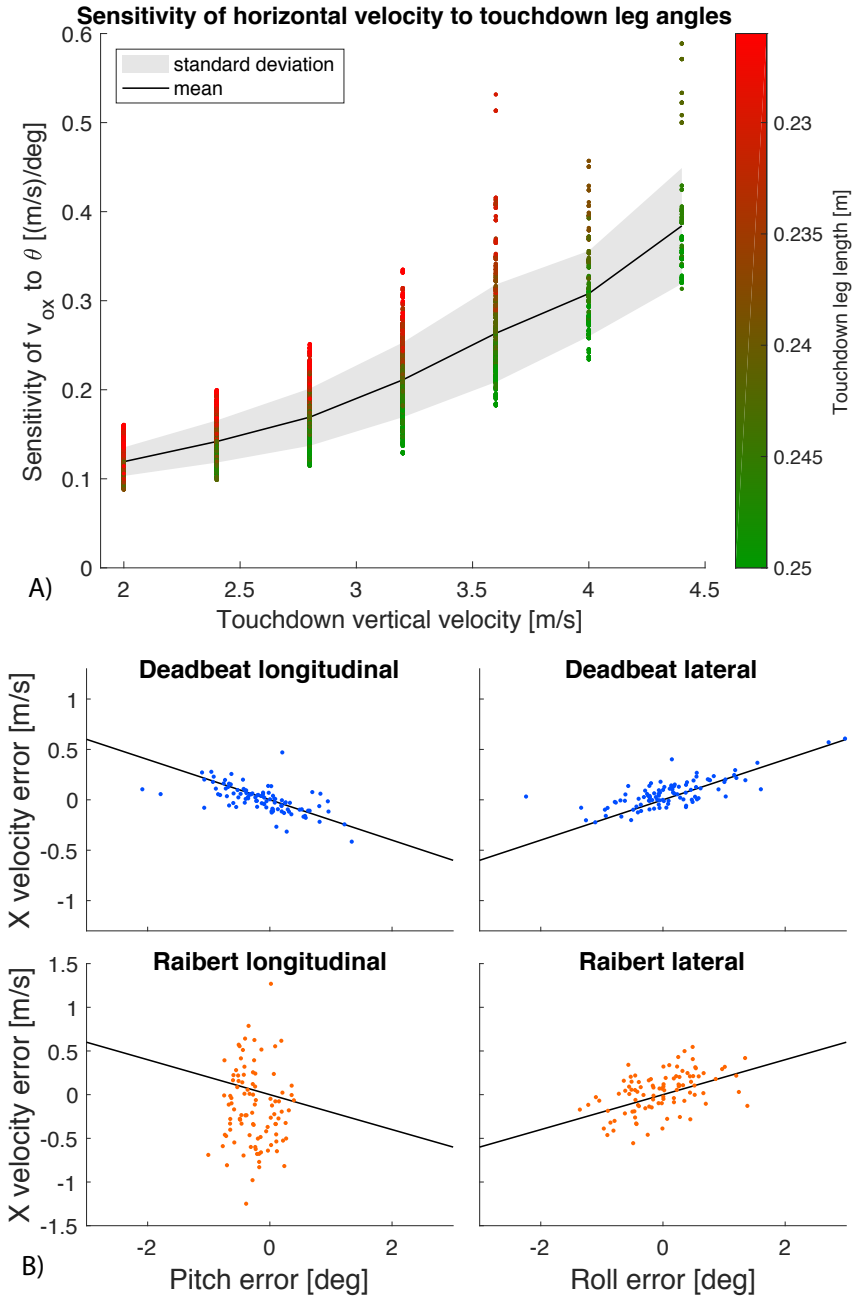


Figure 2.7: A) Simulation data showing the sensitivity of v_{ox} to deviations in θ plotted against $-v_{iz}$ for the entire range of simulated initial conditions. Mean sensitivity is shown in black and one standard deviation is shown in grey. B) Correlation between leg angle error and horizontal velocity error in the random walk experiments. The lines have slope 0.2, as predicted for this task by A).

shown in Fig. 2.7A. Horizontal velocity sensitivity to touchdown leg angles also varies with touchdown leg length l but the effect is smaller than that of vertical velocity.

To investigate how attitude control error propagates to foot placement error, I consider steady hopping over mostly flat terrain in which the vertical velocities change little from hop to hop. On horizontal surfaces, horizontal foot placement error is the product of the takeoff horizontal velocity error and the flight time plus errors due to flight time error. As shown in Fig. 2.7A, the horizontal velocity sensitivity to touchdown leg angles rises somewhat linearly with vertical velocity. Flight time rises approximately linearly with v_{zo} when the apex is significantly higher than the terrain height variation. Therefore the foot placement error rises approximately quadratically with vertical velocity for a given leg angle error. Apex height rises quadratically with increasing v_{zo}, thus foot placement error is approximately linearly related to apex height. Smaller jumps are significantly more precise than higher jumps for a given flight attitude control accuracy. From the numerical sensitivity data, to achieve a foot placement error lower than 0.15 m, the acceptable touchdown leg angle error is 2.8 degrees for $v_{zi} = v_{zo} = 2.4$ m/s but only 0.5 degrees for $v_{zi} = v_{zo} = 4.0$ m/s.

There is also a lower limit to effective hopping height since flight time decreases linearly with takeoff vertical velocity. If the flight time is too short, the flight attitude control will be unable to reorient the leg fast enough and the touchdown attitude error will suffer. Therefore, there is a mid-to-low hop height for which the foot placement accuracy is optimal.

Attitude error propagation sets a foot placement precision limit for a given attitude control accuracy. During the deadbeat velocity controller's random walk task, the touchdown leg angle errors were approximately Gaussian with standard deviations of 0.58 degrees in θ and 0.75 degrees in ϕ. For this task, the numerical sensitivity data predict a takeoff horizontal velocity sensitivity to touchdown leg angle of about 0.2 (m/s)/deg, shown in Fig. 2.7B by black lines. This prediction matches the observed data relatively well. θ error was correlated to v_{ox} error with a coefficient of -0.61 and ϕ error was correlated to v_{oy} error with a coefficient of 0.75 as shown in Fig. 2.7B. This shows that a significant component of the foot placement errors can be attributed to the touchdown leg angle error and the deadbeat velocity controller operated relatively near the foot placement precision limit of the attitude control accuracy.

In comparison, the Raibert controller random walk θ error and v_{ox} error were correlated by a coefficient of -0.04 and the ϕ error and v_{oy} error were correlated by a coefficient of 0.49. The lower correlations, particularly in the longitudinal direction, are because the contribution of touchdown leg angle error to foot placement error is dwarfed by the contributions of other errors in the Raibert controller.

Jumping on a chair and desk

With the deadbeat foot placement hopping controller's superior accuracy, I demonstrated that the robot can jump up on top of a chair and desk as shown in Fig. 2.8. The jumping trajectory and foot placement errors are shown in Fig. 2.9. The plastic folding chair is 0.44 m tall at the front, 0.40 m tall at the back, 0.37 m deep, and 0.38 m wide. Jumping onto

Figure 2.8: The robot jumps up onto a chair and desk (trajectory in blue), then back down (not shown here, but shown in Fig. 2.9).

the chair requires foot placement accuracy better than the horizontal dimensions of the seat in order to avoid falling off. For $v_{iz} = v_{oz} = 3.9$ m/s, the stance initiating the jump onto the chair has an allowable touchdown leg angle error of 0.7 degrees. The desk is a standard 28 inches high (0.71 m), 0.7 m deep, and 1.2 m wide. Both the jump from the ground to the chair and from the chair to the desk are higher than the robot's full body length. The robot jumped 0.5 m longitudinally to mount the chair and then 0.3 m laterally to mount the desk.

This is a challenging task since the chair seat is small in comparison to the foot placement precision from attitude control accuracy; the robot did not always complete it successfully. This task also requires large changes in energy to mount and descend obstacles higher than the robot's size. The robot achieved desired vertical takeoff velocities by adding and removing energy when jumping higher and lower.

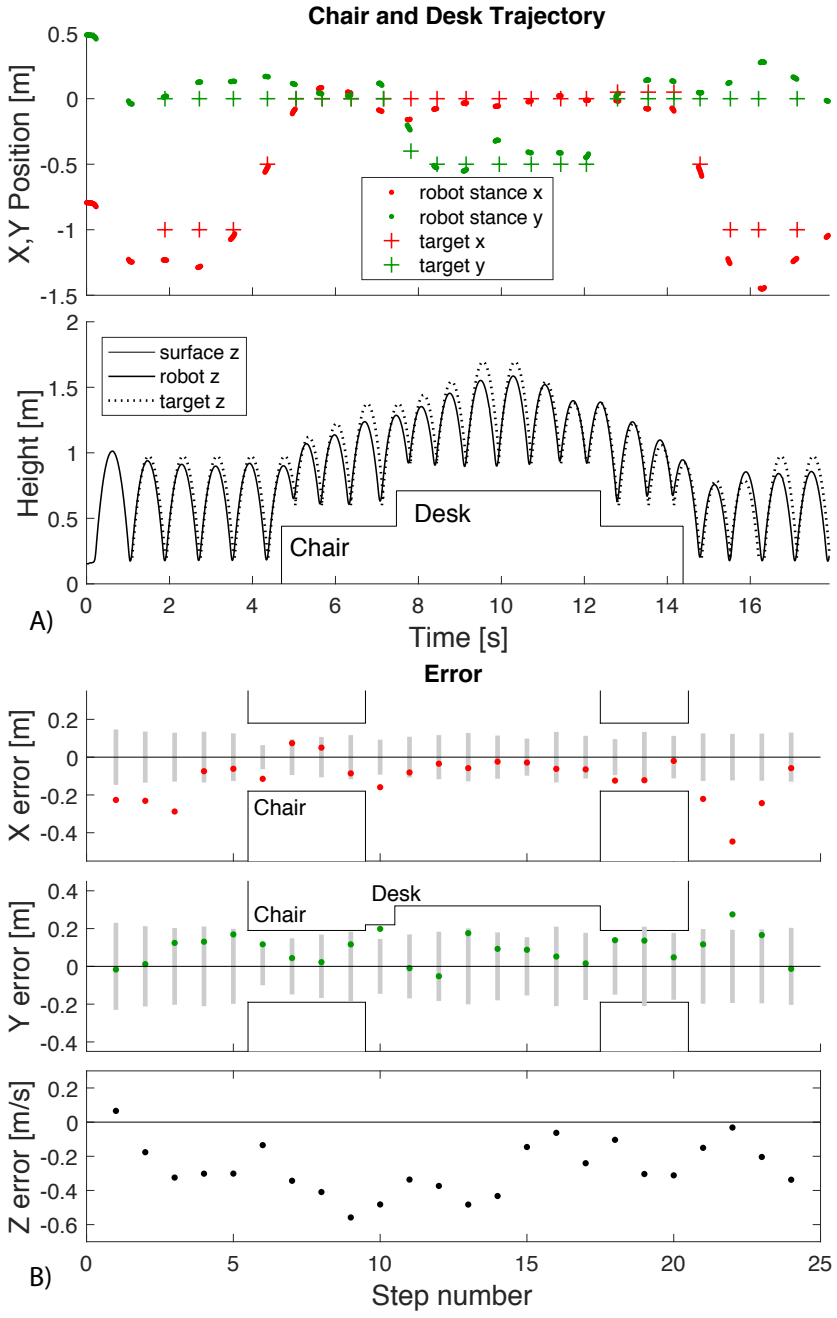


Figure 2.9: A) Foot placements and height of the robot as it jumps onto the chair and desk and then back down. B) Foot placement and v_{oz} error. Grey bars show foot placement error for the standard deviation of the random walk angle error. To avoid falling, foot placement error cannot cross furniture edges.

Jumping on a moving platform

Finally, the deadbeat foot placement hopping controller is quick to compute and is robust to moderate ground disturbances. Due to its quick computation, it is able to quickly retarget to a new desired touchdown location. To demonstrate this, I outfitted a hand-carried wooden board with motion capture markers and commanded the robot to jump to a point on the board. The robot was commanded to alternate between the ground at (0,0,0) and the point on the board. The trajectories of the point on the board and the robot foot placements are plotted in Fig. 2.10.

The robot converged quickly to alternating between (0,0,0) and the board. At nineteen seconds, the platform was lifted off the ground. The robot completed two jumps from the ground to the board before being captured on top of the board at 22 seconds. The robot stabilized on top of the moving board until the board was placed on the ground and moved rapidly to the side at 33 seconds. The robot jumped between (0,0,0) and board twice before the aggressive horizontal jump length caused the robot to slip and fall over.

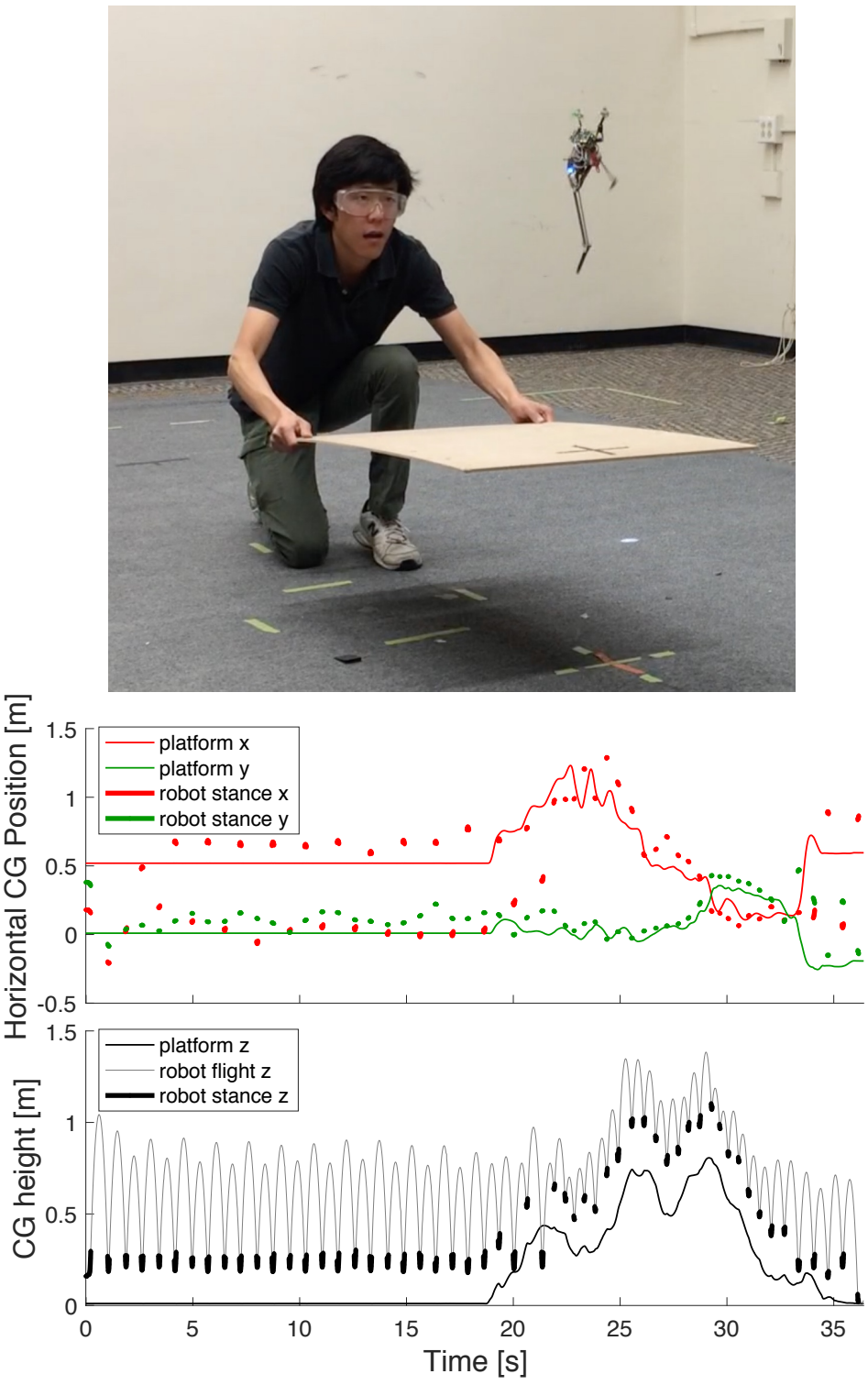


Figure 2.10: Hopping on a moving platform.

2.4 Conclusion

I demonstrate that the Deadbeat Foot Placement Hopping Controller (DFPHC) achieves superior foot placement precision and jumping height adjustment when compared to an optimally tuned Raibert hopper controller for Salto-1P. This controller is capable of rapid changes in direction and velocity including reversing direction and turning at right angles. It is capable of controlling the energy in the system to achieve desired takeoff vertical velocities in a single jump.

I also analyze the sensitivity of the robot's takeoff velocity to touchdown leg angle deflections and show that flight-phase attitude error tolerance becomes tighter for higher hopping in which the vertical velocities are faster. This introduces an interesting tradeoff in jumping strategy. While larger jumps enable locomotion over footholds placed farther apart, foot placement precision degrades quadratically with increasing vertical velocity and linearly with increasing apex height. A larger number of smaller hops enables higher foot placement precision to avoid missing footholds. There is also a lower limit to the vertical velocity and height of small jumps to keep the flight-phase angular velocity within actuator limits. There is an optimal mid-to-low jump height at which the robot achieves its most precise foot placements, though the exact value of this optimal height was not explored in this work.

The deadbeat foot placement hopping controller's high precision and good robustness to ground disturbances enabled it to hop on discontinuous surfaces like office furniture and to track a moving platform. The precision of the deadbeat foot placement hopping controller approaches the precision limit set by the accuracy of the flight-phase attitude control. Future improvements to the accuracy of the flight-phase attitude controller will enable more precise foot placement. Extensions to include the battery state of charge in the controller can improve the vertical velocity and hopping height control.

The third order polynomial deadbeat velocity controller is easy to compute. In chapter 3, a slightly modified version is run by Salto-1P's onboard computation-limited embedded processor. In chapter 4, a launch-and-land strategy demonstrates a way to improve over SLIP-like hopping accuracy. Future work can investigate a more theoretically founded understanding of hopping control sensitivity to errors based in simple mathematical models like SLIP, stance-phase strategies that could correct touchdown errors, and the effects of non-rigid or sloped terrain on jumping performance and control strategy.

Chapter 3

Hopping Estimation

3.1 Introduction

Motivation

Chaining together large, fast jumps allows a locomotor to quickly clear obstacles and gaps using sloped or widely separated footholds. However, robots capable of continuous high hopping with longer flight times than stance times either rely on external sensing and control or are constrained by rigs like booms, planes, or umbilicals [58] [86] [67] [9]. In chapter 2, Salto-1P relied on external motion capture control for pose and velocity estimation in order to execute its precise jumps. As shown in chapter 2, precise attitude control and estimation is critical for accurate jumping, hence the requirement for precise external sensors or constraints. An estimation scheme for hopping using only onboard sensing and processing would be useful for freeing high-performance hopping robots from the supporting laboratory equipment.

In this work, my collaborators and I develop an onboard SLIP Hopping Orientation and Velocity Estimator (SHOVE) for the Salto-1P small jumping robot to achieve stable hopping without the support of external sensing or processing. [1]

Prior Work

Significant prior work has investigated the dynamics and control of running and hopping. We draw on the Spring Loaded Inverted Pendulum (SLIP) [12] and adopt the framework of Raibert's controller by selecting touchdown leg angles to direct a SLIP-like hopping robot [58]. This flight-phase control is effective even for jumping robots with short impulse-like stance phases [32].

Batts et al. demonstrated one of the first SLIP-like monopeds that was both power-autonomous and estimated its own state with onboard sensing alone [9]. It completed 19 hops in 7 seconds but was not stable indefinitely.

Integration of triaxial rate gyroscopes is a common attitude estimation solution. However, integrated angular velocities will eventually drift as errors from the rate gyroscope accumulate. A correction is required to counter this drift.

In Unmanned Aerial Vehicles (UAVs), accelerometer or inclinometer measurements of the gravity vector can serve as a reference. Kalman filters [48] and complementary filters [8] are common methods to fuse these measurements. Spacecraft frequently use some flavor of Kalman filter to fuse attitude sensors like sun or star trackers into the attitude estimate [46]. These sensors require mostly unobstructed views to their known references and would not be appropriate for a terrestrial robot in a cluttered environment.

[1] This chapter is revised from work presented as "Drift-free roll and pitch estimation for high-acceleration hopping," *IEEE International Conference on Robotics and Automation (ICRA)*, 2019 [83]. J. K. Yim was the primary author. E. K. Wang developed the protective shell and robot calibration and assisted in robot experiments. This material is based upon work supported by Army Research Office Grant No. W911NF-18-1-0038.

Figure 3.1: Salto-1P jumps outdoors without external sensing.

Walking platforms that do not have a flight phase and remain dynamically anchored to the ground possess low acceleration periods that can be used to counter gyro drift [60]. Walking humans have been similarly tracked with shoe-mounted inertial sensors [26].

Running robots sometimes use other sensors to augment rate gyroscopes like an infrared sensor pointed at the ground [51]. Others rely on low gyroscope bias so that the drift is small over their operating duration [76].

High-power jumping robots possess neither gravitational references in their free-fall flight nor low stance accelerations in which to correct gyro integration. Salto-1P spends a mere 70 ms in stance phase, during which it experiences a sustained 14 g acceleration as well as large vibrations induced by the high-impact landing, making accelerometer measurement of a gravity vector very difficult. While magnetometers are frequently used to prevent heading drift, a gravity vector measurement is not available. Although highly precise rate gyroscopes like fiber optic gyroscopes minimize drift, they are still subject to limits on maximum measurable angular rate. In saltatorial locomotion, shocks can saturate a gyroscope and cause angle estimate error that must be corrected to maintain stable hopping. Thus, we develop SHOVE, a solution that estimates and controls robot velocity and angle for a SLIP-like monopedal hopping robot with impulse-like stance phases. SHOVE operates by using 1) stance velocity estimation through an improved dynamic model compared to chapter 2 and subsequently 2) attitude estimate correction from velocity control via touchdown attitude.

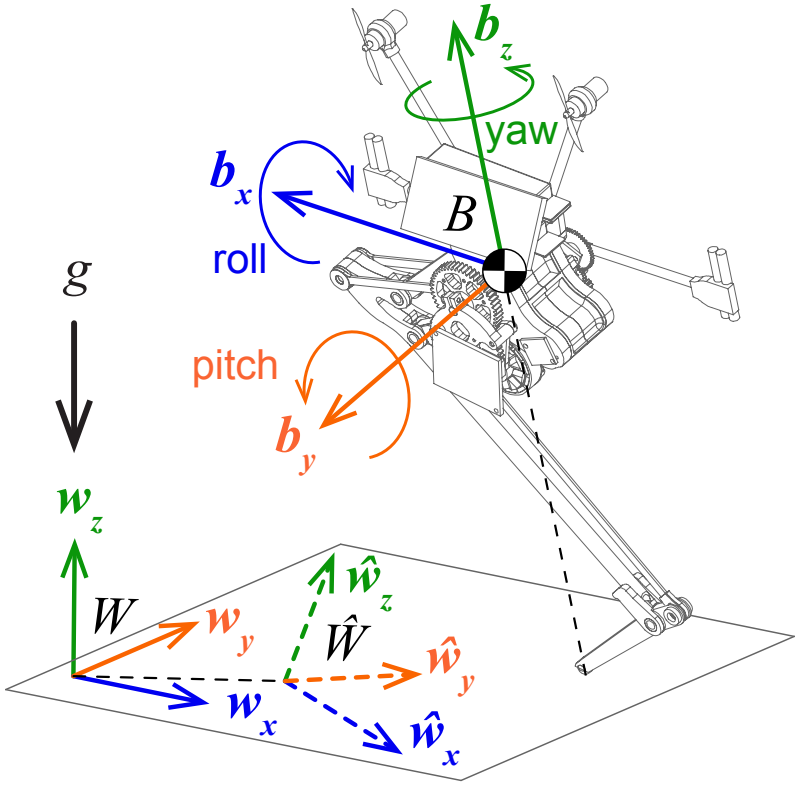


Figure 3.2: Robot body, world, and estimated world reference frames.

3.2 Methods

Jumping motion can be divided into alternating stance phases (stances) and flight phases (flights). Touchdown (TD) is the transition between flight and stance when the foot strikes the ground. Liftoff (LO) is the transition between stance and flight when the foot leaves the ground.

In flight, a jumping robot has little control over its center of gravity (CG) trajectory without specialized means to apply large forces in the air. Neglecting drag, its CG trajectory in flight follows a ballistic parabola.

The robot estimates its orientation relative to a world-fixed reference frame W with basis $\boldsymbol{w}_x$, $\boldsymbol{w}_y$, $\boldsymbol{w}_z$ as shown in Fig. 3.2. The robot can equivalently be considered to be estimating the world frame's orientation relative to the robot's frame B. We denote vectors in the estimated world frame, $\hat{W}$, with a hat.

SLIP Hopping Orientation and Velocity Estimator

The following sections introduce our SLIP Hopping Orientation and Velocity Estimator (SHOVE). This estimation scheme attempts to correct roll and pitch attitude errors imme-

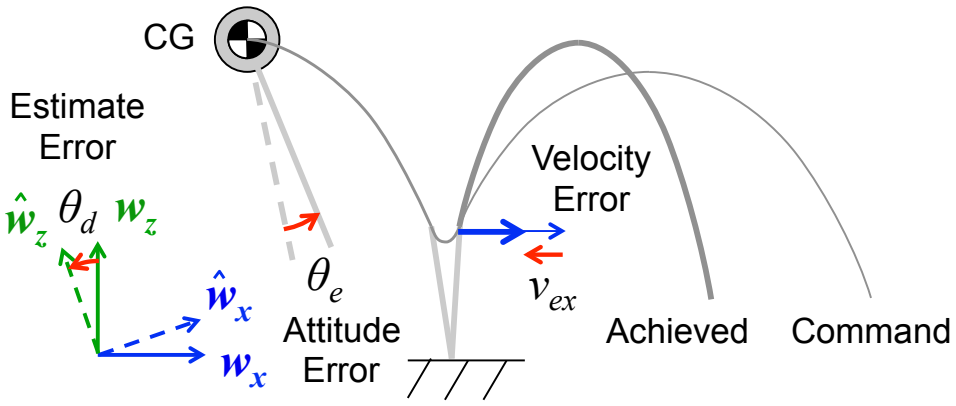


Figure 3.3: Attitude errors propagate to liftoff velocity errors so liftoff velocity can be used as an attitude correction signal.

diately following LO. When the robot detects LO, it makes the following computations:

1. Calculate liftoff velocity using model of leg dynamics
2. Compute liftoff velocity error by comparing expected velocity from controlled touchdown attitude with 1)
3. Compute attitude corrector values for roll and pitch using liftoff velocity error
4. Subtract correction values from attitude estimate

Attitude Corrector

This section describes our attitude corrector algorithm for the roll and pitch of SLIP-like hopping robots. Without exteroceptive sensors like cameras or rangefinders, a jumping robot in flight has little indication of the direction of gravity or if its attitude estimate $\hat{W}$ agrees with the world frame W. However, attitude errors become apparent in stance since the TD and LO velocities will differ from predictions that use the erroneous $\hat{W}$ attitude estimate. For example, the horizontal TD and LO velocities should be equal if drag is negligible; however, if $\hat{\boldsymbol{w}}_z$ is deflected from $\boldsymbol{w}_z$, then the horizontal TD and LO velocities will appear to differ. Using relationships like these, an attitude corrector can cancel attitude errors that deflect $\hat{\boldsymbol{w}}_z$. While this attitude correction works for any angle parameterization, we use ZXY Euler angles (yaw ψ, roll ϕ, pitch θ) for convenience. In this parameterization, the attitude corrector estimates the pitch error θ_d and roll error ϕ_d, respectively, from $\hat{W}$ to W.

A large class of hopping controllers for SLIP-like robots set TD leg angles like θ and ϕ to regulate the LO velocity vector $\boldsymbol{v}_{LO}$ such that it follows a commanded LO velocity vector $\boldsymbol{v}_c$. Among these controllers are Raibert's seminal controller [58] and many proposed subsequently, such as [66, 78, 67], and the DFPHC in Chapter 2 [81].

For these controllers, an attitude error about a horizontal axis (like θ_d and ϕ_d) will cause a LO velocity error $\boldsymbol{v}_e = \boldsymbol{v}_{LO} - \boldsymbol{v_c}$ as investigated in Chapter 2 and illustrated in Fig. 3.3. The x component of LO velocity $\boldsymbol{v}_{LOx}$ depends most strongly on θ and $\boldsymbol{v}_{LOy}$ depends most strongly on ϕ. These relationships between angle and horizontal velocity are relatively close to independent and linear as shown in Chapter 2.

This relationship between attitude error and horizontal LO velocity means that horizontal LO velocity errors v_{ex} and v_{ey} can be used to estimate the error between the achieved TD angles, θ_{TD} and ϕ_{TD}, and those commanded by the velocity control, θ_c and ϕ_c :

$$K_{vx} v_{ex} \approx \theta_e = \theta_{TD} - \theta_c$$
$$-K_{vy} v_{ey} \approx \phi_e = \phi_{TD} - \phi_c$$

where K_{vx} and K_{vy} are the coefficients relating angle error to horizontal velocity. From Chapter 2, K_{vx} and K_{vy} are approximately 3 degrees per m/s for Salto-1P.

Since the onboard attitude controller operates using the onboard estimates $\hat{\theta}$ and $\hat{\phi}$, these above expressions can be expanded to be in terms of estimator errors θ_d and ϕ_d. SHOVE assumes θ_d and ϕ_d are small so that $\hat{\boldsymbol{v}}_e \approx \boldsymbol{v}_e$ and $\hat{\boldsymbol{v}}_{LO} \approx \boldsymbol{v}_{LO}$. This assumption is acceptable since variations in v_{LOx} and v_{LOy} caused by θ_e and ϕ_e respectively are larger than those in $\hat{\boldsymbol{v}}_{LO} - \boldsymbol{v}_{LO}$ caused by θ_d and ϕ_d for small angles.

$$K_{vx} \hat{v}_{ex} \approx \hat{\theta}_{TD} - \theta_d - \theta_c$$
$$-K_{vy} \hat{v}_{ey} \approx \hat{\phi}_{TD} - \phi_d - \phi_c$$

Isolating θ_d and ϕ_d and multiplying each term by a gain produces the SHOVE attitude correction terms θ_s and ϕ_s which are computed using onboard measurements and added to $\hat{\theta}$ and $\hat{\phi}$ estimates immediately following LO:

$$\theta_s = K_x \hat{v}_{ex} - K_e(\hat{\theta}_{TD} - \theta_c)$$
$$\phi_s = -K_y \hat{v}_{ey} + K_e(\hat{\phi}_{TD} - \phi_c)$$

K_x and K_y are the feedback gains that correct the attitude error based on horizontal velocity error. If $K_x = K_{vx}$ and $K_y = K_{vy}$, the estimator is approximately deadbeat, but may be over-aggressive. Reducing K_x and K_y makes the estimator less aggressive. K_e accounts for estimated controller error and should be between 0 and 1. The experimentally tuned gains are $K_x = 1.5$ degrees per m/s and $K_y = 1.0$ degrees per m/s. We use $K_e = 0.5$ in both directions.

This estimation scheme assumes that drag is negligible so that horizontal velocity at LO and TD are close. Motion capture velocity measurements indicate drag is less than 0.07 N (less than 7% of bodymass) even at the highest velocities. The estimation also assumes that the velocity controller that selects TD leg angles is relatively accurate and ψ error changes little (only a few degrees) in one jump. These are both true of the controller developed in Chapter 2. In the results we show experimental evidence that this attitude estimation scheme runs satisfactorily in real conditions and enables the fully-autonomous operation of a monopedal hopping robot without the support of any exernal sensing or computation.

Velocity Estimation

Both the preceding attitude corrector and the following velocity control require accurate estimates of the robot's velocity. Since flight is ballistic, flight velocity can be easily computed from LO velocity at the end of stance. Stance velocity estimation is the second component of SHOVE.

Since Salto-1P uses a straight-line linkage for its leg, stance CG velocity can be computed from leg length, leg extension velocity, and angular velocity. In order to estimate the velocity of its CG along its leg axis $\boldsymbol{b}_z$ in stance, Salto-1P uses a simple model of its leg extension dynamics with two state variables: leg length l and leg velocity $\dot{l}$. The observer is updated with measurements from the leg linkage encoder θ_{leg} and motor encoder θ_{motor} at 500 Hz. The observer approximates the kinematic relationships between the leg extension, crank rotation, and mechanical advantage (MA) using lookup tables l_{enc}, θ_{crank}, and MA indexed by the leg linkage encoder. f_{spring} is a quadratic nonlinear spring model and K_f is a dry friction coefficient. The model is:

$$\begin{aligned}
\theta_{spring} &= \theta_{crank}[\theta_{leg}] - \frac{1}{\text{gear}}\theta_{motor} \\
F_{foot} &= MA[\theta_{leg}]\,(1 - K_f)\,f_{spring}(\theta_{spring}) \\
e &= l_{enc}[\theta_{leg}] - l[t] \\
l[t+1] &= l[t] + \dot{l}[t]\Delta t + K_l e \\
\dot{l}[t+1] &= \dot{l}[t] + \left(\frac{F_{foot}}{m} + g\right)\Delta t + K_{\dot{l}} e
\end{aligned}$$

where K_l and $K_{\dot{l}}$ are the observer gains. This model neglects forces perpendicular to the leg and centrifugal force.

Salto-1P detects LO when $\theta_{spring} = 0$ or l reaches maximum extension. At LO, Salto-1P estimates $\boldsymbol{v}_{LO}$

$$\boldsymbol{v}_{LO} = R[0, 0, \dot{l}]^\top + R(\boldsymbol{\omega} \times [0, 0, l]^\top) + [v_{bx}, v_{by}, 0]^\top$$

where R is the rotation matrix from B to $\hat{W}$ and $\boldsymbol{\omega}$ is the angular velocity measured by the IMU rate gyroscopes. Small experimentally-tuned bias terms v_{bx} and v_{by} added to the LO horizontal velocities compensate for small mechanical construction asymmetries.

In flight, drag is neglected in computing the velocity estimate $\hat{\boldsymbol{v}}$. $\hat{v}_x$ and $\hat{v}_y$ are unchanged while the vertical component $\hat{v}_z$ decelerates by g. Salto-1P detects TD when its spring deflection exceeds a threshold. At TD, Salto-1P initializes $\dot{l} = \hat{v}_z$ and begins stance velocity estimation again.

We experimentally measured the leg mechanics on a test stand shown in Fig. 3.4 to identify parameters for l_{enc}, θ_{crank}, MA, f_{spring} and K_f. By pushing a weighted cart along a vertical linear slide while mounted on a test-stand load cell, Salto-1P mimicked jumping under its own mass while its body remained fixed. We fit a third order polynomial for l_{enc} and a fifth order polynomial for θ_{crank} using image tracking for foot position and onboard

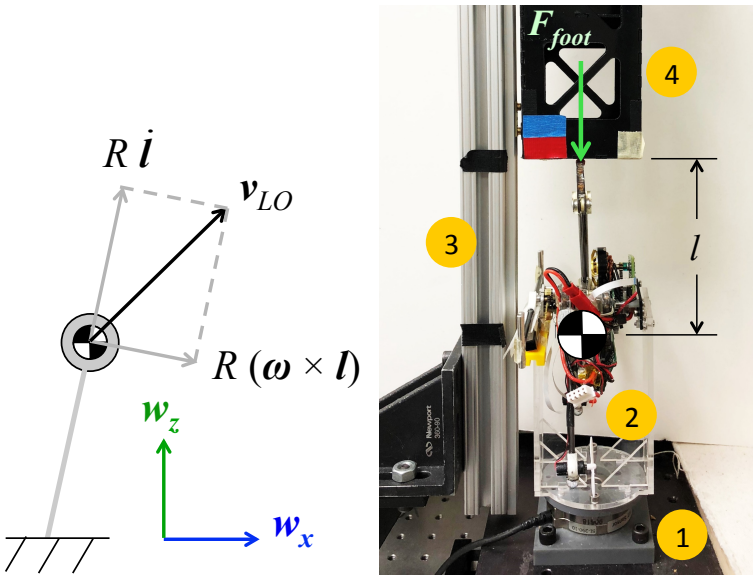


Figure 3.4: Left: Calculation of liftoff velocity. Right: Calibration stand: 1) load cell, 2) robot mounted upside-down in load cell carriage, 3) linear slide, 4) cart with red & blue fiducial for tracking.

encoders for femur and motor angles. We derived MA from l_{enc} and θ_{crank}. To save onboard computation, l_{enc}, θ_{crank}, and MA were discretized into lookup tables. From the test-stand's load cell we fit f_{spring} and K_f.

Velocity Control

Salto-1P controls its hopping height and horizontal velocity using a modified version of the deadbeat foot placement hopping controller developed in Chapter 2. This controller selects commanded TD leg retraction l_c, roll angle ϕ_c, and pitch angle θ_c using the robot's onboard estimate of its velocity in flight $\hat{\boldsymbol{v}}$ and the commanded LO velocities $\boldsymbol{v}_c$.

Since $\hat{v}_z$ changes over time, the deadbeat controller commands also change as the robot falls through flight. Similarly to [78], this command trajectory achieves $\boldsymbol{v}_c$ from whatever point the foot contacts the ground in its post-apex flight path.

Proportional derivative (PD) controllers command the tail and thruster attitude actuators and leg motor to achieve the commanded TD attitude and leg length during flight.

Hardware Platform

We demonstrate SHOVE on the Salto-1P platform developed in [32]. Salto-1P is a 0.10 kg, 0.31 m tall monopedal hopping robot built around a series-elastic power modulating leg. In the air, a balanced inertial tail and two small lateral propellers control its attitude. Salto-1P

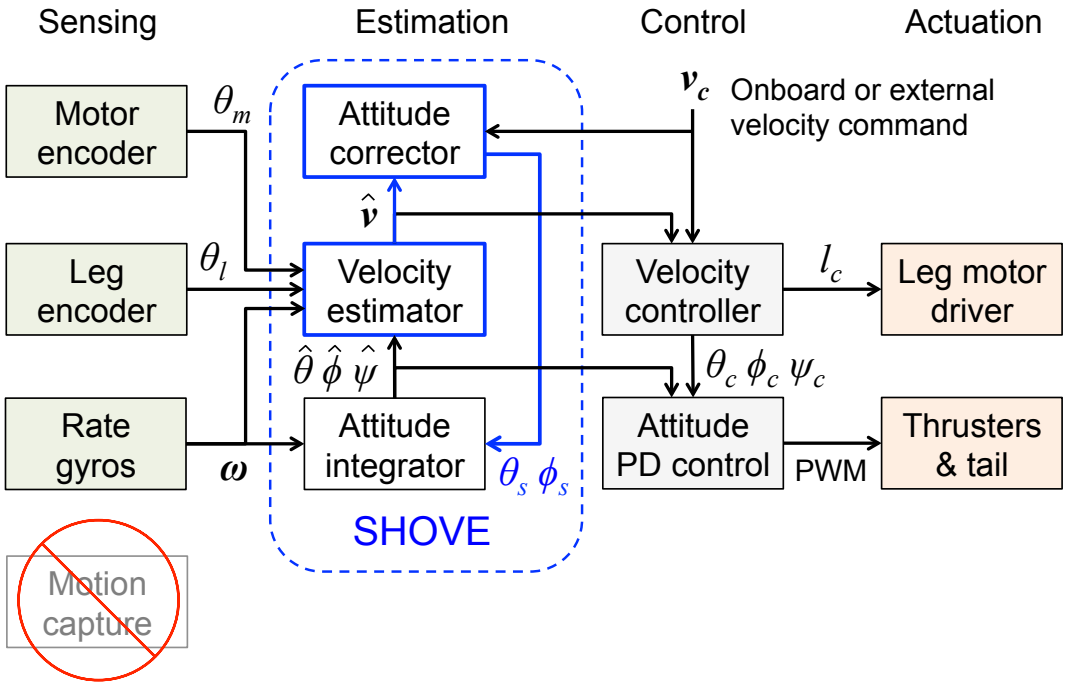


Figure 3.5: Robot system block diagram.

has an MPU 6000 TDK InvenSense 6-axis IMU. Its rate gyroscopes are set to a sensitivity of ± 2000 deg/s and its accelerometers are set to a sensitivity of ± 16 gs, both measured as 16 bit signed integers. Three 14-bit magnetic encoders measure leg, motor, and tail angle. Salto-1P also has memory for logging experimental data, and an XBee radio for communication.

Fig. 3.5 depicts the interaction of the estimation and control systems described in the above sections with the sensors and actuators on Salto-1P.

Salto-1P parameterizes its rotation with ZXY Euler angles (yaw ψ, roll ϕ, pitch θ). This parameterization experiences singularities at $\phi = \pm 90°$, near which the robot never operates. IMU rate gyroscope angular velocity measurements update the Euler angles by Reimann integration using Bhaskara I's cosine approximation [31] to reduce computational load.

Salto-1P is equipped with a 4 gram shell for operation outside a laboratory environment. The shell is made of a thermoformed 380 μm polycarbonate sheet and laser-cut delrin. The casing encompasses the microcontroller board, motor driver board, and gear box which would otherwise be easily damaged by outdoor surfaces.

Initial Attitude Estimate

Before it begins jumping, Salto-1P initializes its attitude estimate in a two-step process. In its starting pose, Salto-1P rests on three points: its toe, rear tarsus "ankle" joint, and one end of its tail. Since the robot is initially stationary, the roll and pitch estimates are first

set by the accelerometer readings. Initial yaw is arbitrarily set to zero. To refine its attitude estimate, the robot stands up on its toe before jumping. The robot activates it thrusters and commands the roll PD control to zero the roll angle. The control action of the thrusters is added to the roll estimate so that the roll estimate stabilizes at zero when the robot's roll angular velocity and thruster action are both zero. Once roll is balanced, the tail activates to pitch the robot off of its ankle. The control action of the tail is similarly fed back to the pitch estimate. Once the robot has stabilized, its attitude estimate is zero in roll and pitch and the robot's CG is directly above its toe. This allows the initial attitude estimate to ignore small angle errors in the mounting of the IMU.

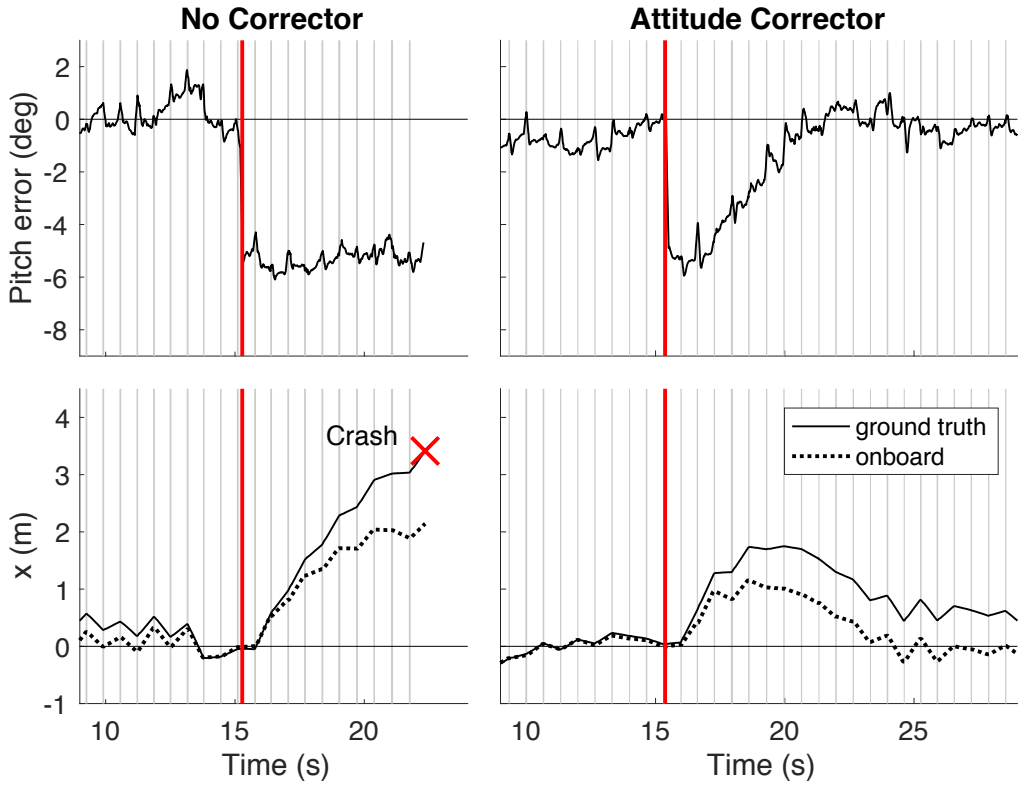


Figure 3.6: Disturbance Experiment. At 15.3 seconds (orange line), a software command inserts 5 degrees of pitch error. Without attitude correction, the robot fails to recover and crashes into a wall. With the attitude corrector, the robot's attitude estimate recovers in 10 jumps. Stance phases are shown in grey.

3.3 Results

Attitude Corrector

To test the efficacy of the SHOVE attitude corrector, we observed its response to a simulated attitude estimate disturbance injected by a software command. This involved two experiments: one with the attitude corrector enabled, and one with it disabled so that the robot relied only on rate gyroscope integration. The robot attempted to hop in place at $x = 0, y = 0$ using only onboard estimation and the velocity control policy $v_{cx} = -x$ and $v_{cy} = -y$. It estimated its position by dead-reckoning, integrating its velocity estimates over time. Motion capture recording provided ground truth measurements.

At 15.3 seconds, a radio command injected -5 degrees of pitch error into the attitude estimate, shown in Fig. 3.6. Without the attitude corrector, the robot pitch error remained offset by -5 degrees. This error caused the robot to jump forwards until it collided with the wall of the laboratory. With the attitude corrector active, the robot recovered its attitude estimate in 10 jumps and did not advance more than 2 m away from its starting position using onboard dead-reckoning. The attitude error caused liftoff velocity error which was integrated into a position estimate error. The position estimate drifted behind the ground truth after the attitude disturbance, but the drift rate halted once the attitude corrector adjusted the attitude estimate.

Dead Reckoning

To test the long-term stability of SHOVE attitude estimation, the robot followed a preprogrammed rectangular trajectory 2 m long and 1 m wide using onboard dead-reckoning and no external inputs. The robot was commanded to hop in place at one corner of the rectangle for 4 seconds, then at 0.5 m/s forwards and backwards along the 2 m legs of the rectangle and at 0.25 m/s left and right along the 1 m legs of the rectangle. This trajectory repeated every 20 seconds. Commanded horizontal velocities followed the control policy $v_{cx} = x_c - x + v_{rx}$ and $v_{cy} = y_c - y + v_{ry}$ where (x_c, y_c) was the currently desired point along the rectangular trajectory and (v_{rx}, v_{ry}) was the desired velocity along the rectangle. The commanded liftoff vertical velocity remained constant at $v_{cz} = 2.5$ m/s. This corresponded to a jump of 0.31 m, or about one body-length of the fully extended robot. Onboard estimates were again compared to motion capture.

The robot hopped stably for longer than the 200 second data collection period. During this period, it completed just over 300 jumps and completed 10 cycles of the rectangular path. Although the robot was commanded to jump with a vertical component of 2.5 m/s, steady state controller error resulted in jumps averaging 2.83 m/s vertically. Attitude error in pitch and roll had standard deviations of 1.10 degrees and 0.91 degrees respectively. While yaw drifted slightly without correction under raw integration, the roll and pitch angles did not significantly drift as shown in Fig. 3.7. The dead-reckoned onboard position estimate

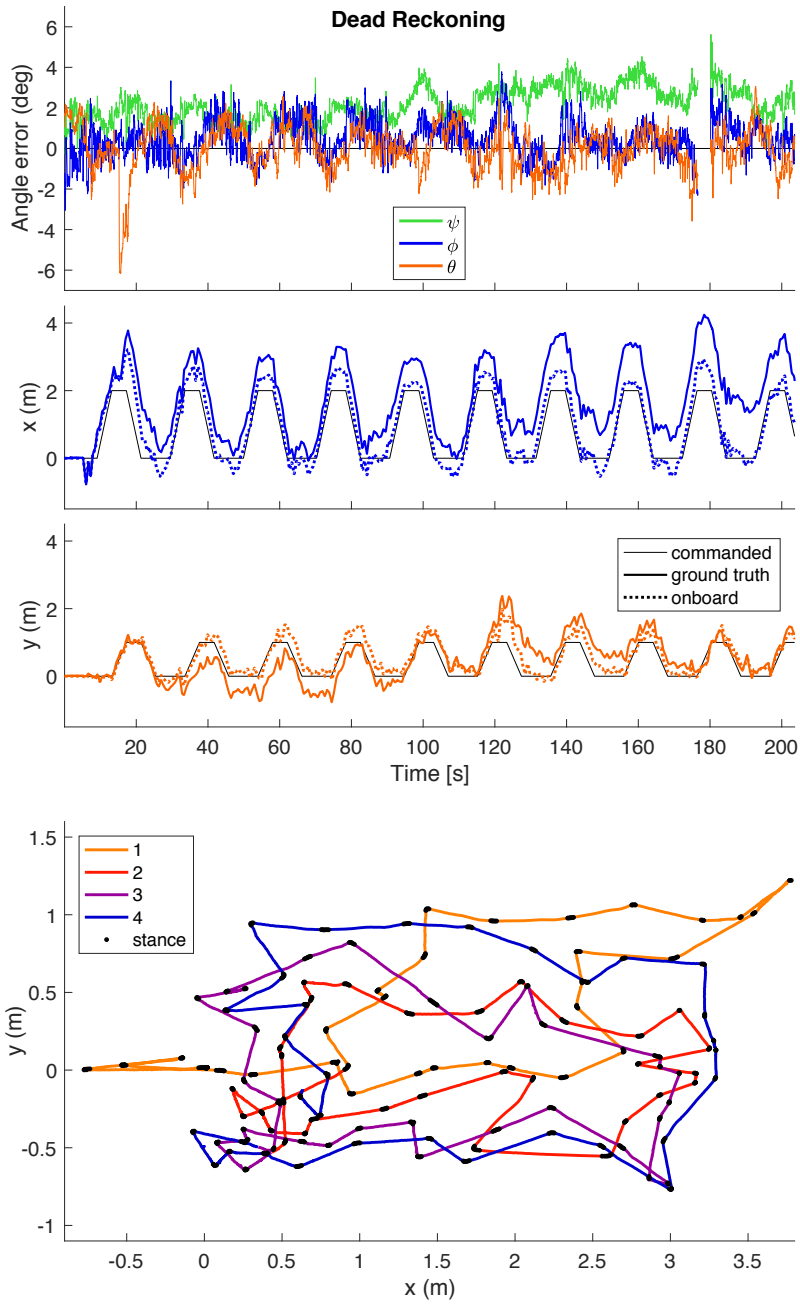


Figure 3.7: Dead reckoning experiment hopping along a 2 m by 1 m rectangle. Top: Attitude error. Yaw drifts slightly while roll and pitch remain near 0. The robot exited the motion capture tracking region at 176.7 seconds and re-entered it at 180.1 s. Middle: robot x and y position compared to onboard estimate and command. Bottom: Overlay of first four consecutive rectangles showing position drift.

drifted less than 2 m away from the ground truth during the 200 second run, an average net displacement of less than 1 cm every second.

This performance demonstrates that the robot can hop stably under onboard SHOVE attitude and velocity estimation. The drift is low enough that low-rate external sensing from auxiliary sensors or guidance would be sufficient to direct the robot to desired positions.

Attitude Estimation on Compliant Terrain

The liftoff velocity estimate relies on accurate estimates of the leg velocity. The velocity estimator assumes that the foot is fixed at the touchdown point on the ground during stance so that the CG velocity can be simply computed from the velocity and angular velocity of the leg. However, compliant terrain deforms under the foot when landed on. Furthermore, terrain compliance can reduce the robot's jump energy and bias its velocity control.

In this experiment, the robot hopped on foam to test how much terrain compliance would disturb the SHOVE attitude estimation and hopping control. Since Salto-1P sinks too far into the foam and cannot stand on its toe to initialize its attitude estimate, the robot initialized on a wooden board placed on top of the foam. This board was quickly removed once the robot made its first leap. This test was compared to a control test on the more rigid carpeted concrete floor of the motion capture room on which the previous two experiments were run. For both of these tests, the motion capture system provided velocity commands $v_{cx} = -x$ and $v_{cy} = -y$ to ensure that the robot remained on the foam and did not drift off. The vertical velocity was commanded to vary sinusoidally over time according to $v_{cz} = 3 + 0.5\cos(t)$ to produce a range of jumping heights. The resulting velocities and attitude errors are shown for both experiments in Fig. 3.8.

The robot performed well on foam despite a spring constant for small deflections of 170 N/m (tested with a 5 mm diameter rod similar to Salto-1P's spherical toe) and significant damping. This stiffness is significant in comparison to Salto-1P's effective leg stiffness of approximately 140 N/m at touchdown.

While the robot underestimated its liftoff velocity by an average of 0.33 m/s on the foam, the estimator remained stable and the robot maintained its hopping position.

Human Control

We set up joysticks for human control of Salto-1P's hopping. Two joysticks independently command horizontal x and y velocities, yaw rate, and vertical velocity at liftoff. The commanded horizontal velocity is limited when the commanded vertical velocity is low in order to avoid causing the robot to stumble.

With joystick commands, Salto-1P can operate outside the laboratory without additional offboard sensing. Human direction was sufficient for the robot to hop on flat surfaces and surmount an obstacle higher than the robot's full body length. In Fig. 3.9, the human operator directed the robot to jump in place, and then advance towards and onto a step 0.43

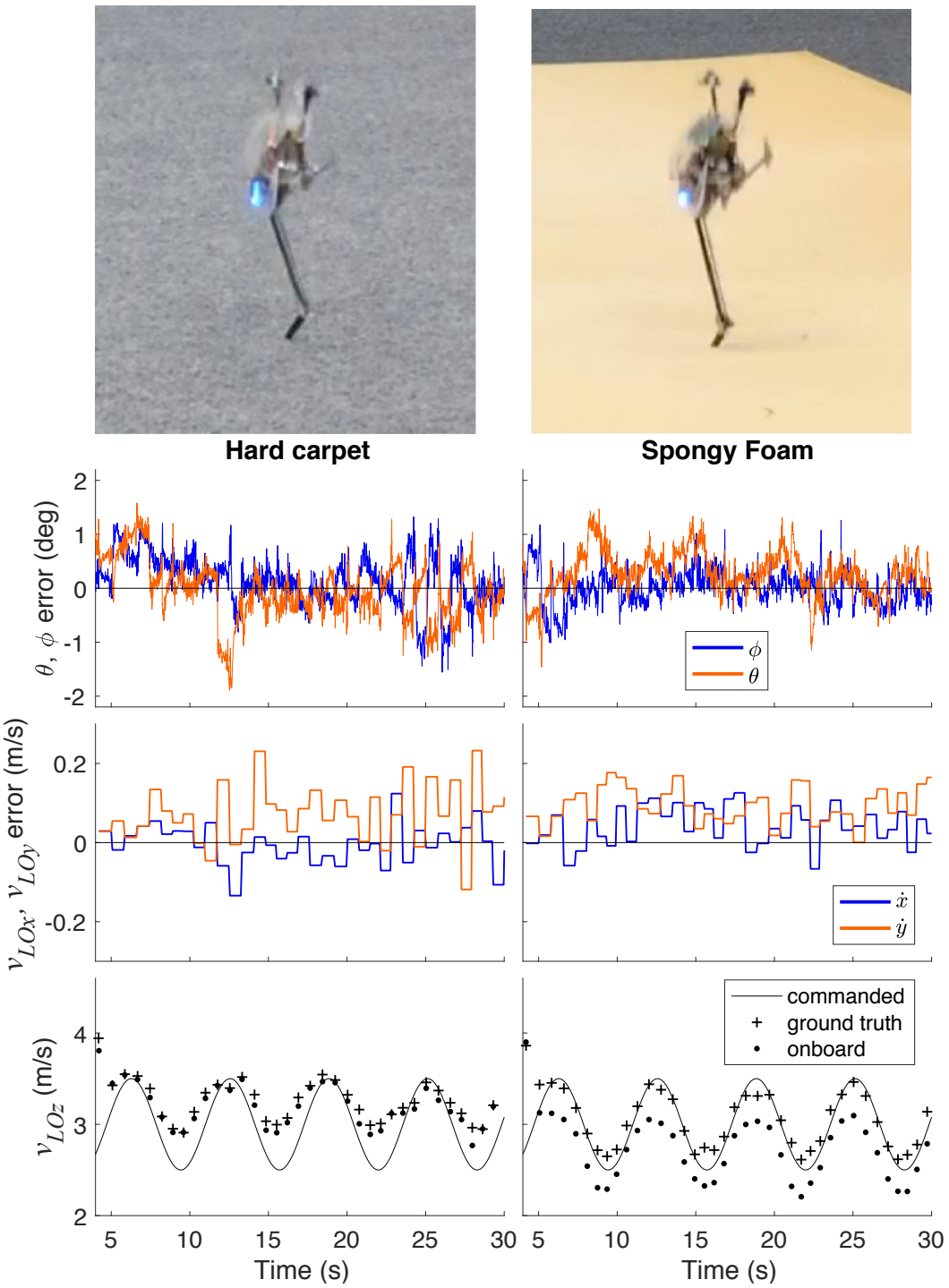


Figure 3.8: Experimental comparison on rigid and spongy terrain.

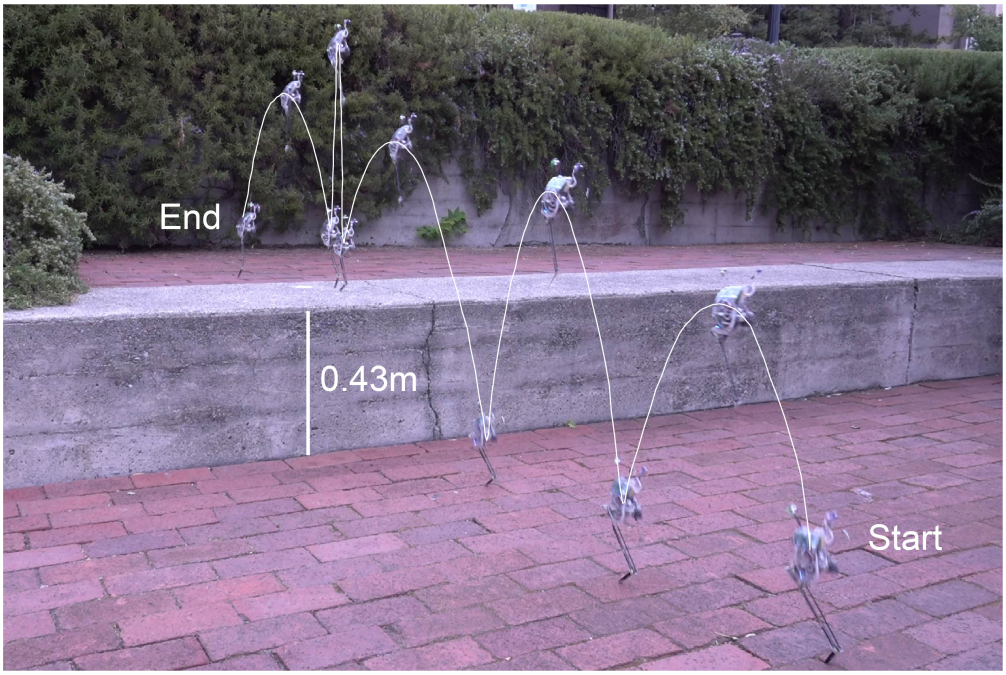


Figure 3.9: Fully onboard run under human directed $\boldsymbol{v}_c$.

m high. The robot was then directed to hop to the right and left on top of the step. This run lasted 19 seconds before human error directed the robot into a bush on the left.

3.4 Conclusion

We develop an attitude and velocity estimation scheme, SLIP Hopping Orientation and Velocity Estimator (SHOVE), for SLIP-like robots and demonstrate it on a small jumping robot, Salto-1P. We also enable human control using joysticks. Together, these systems enable the operation of a monopedal hopping robot outdoors and without the support of external sensing or processing. With human guidance, Salto-1P is able to navigate environments with features taller than its bodylength. The processing load is low enough that it can run at 500Hz on Salto-1P's onboard dsPIC processor.

We demonstrate that the estimator performs stably even when it encounters compliant terrain. It is also able to recover from attitude estimate disturbances that would not be recoverable for angular rate gyroscope integration alone.

This system is subject to several limitations. Although the estimator can compensate for small angle errors in the IMU mounting, the velocity estimation still requires sensitive hand-tuning to compensate for some offsets in liftoff velocity. The attitude estimate errors are on the order of about a degree, causing the robot's foot placements to scatter about a half meter from jump to jump. Foot placements are not yet accurate enough for the robot to use its onboard estimation to execute maneuvers like climbing stairs without significant chance of colliding with a step edge.

Augmentation with additional onboard or offboard exteroceptive senses like vision or distance measurements and associated greater processing power could improve estimation accuracy in favorable conditions. SHOVE estimates could be fused with these sensing modalities, or in conditions that impede other sensors like darkness, fog, dust, or communications failure, SHOVE could operate alone as a fallback.

Future work includes theoretical investigation of the estimator stability as well as improved filtering and estimation to improve accuracy and precision. Higher precision estimation and control can enable jumping on more finely varied surfaces like stairs, furniture, or other outcroppings. Investigations into interactions with terrain compliance will aim to enable consistent estimator and control performance even on soft substrates like upholstery or natural foliage. Improved estimation of robot dynamics can enable both investigation of terrain properties and diagnostics to determine robot health as its mechanisms age.

The next chapter explores a very different monopedal hopping motion consisting of isolated hops separated by balanced stance phases in contrast to the bouncing motion explored in this chapter and the preceding chapter 2.

Chapter 4

Launching and Landing

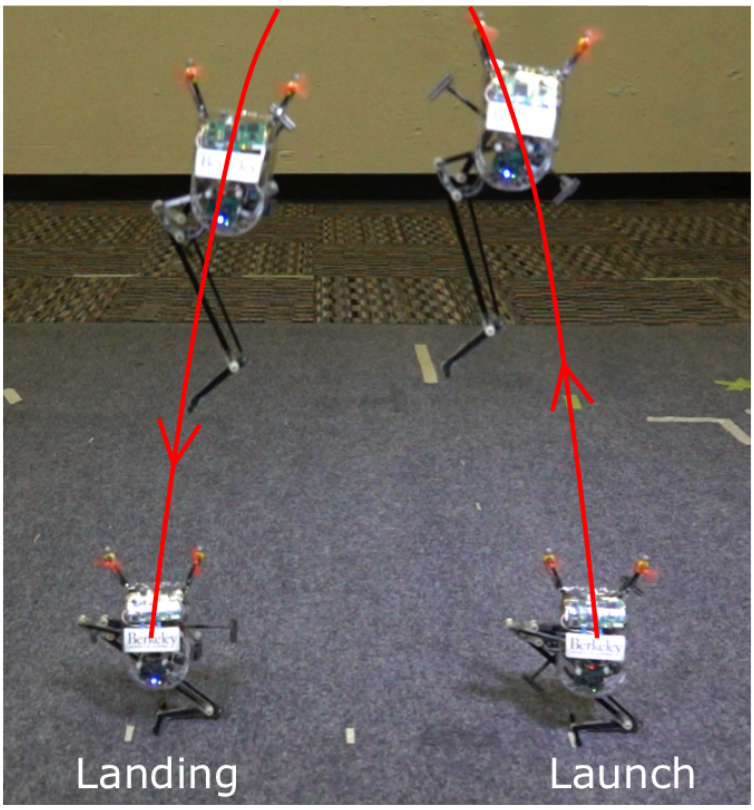


Figure 4.1: Salto-1P leaps and lands.

4.1 Introduction

Chapters 2 and 3 developed control and estimation for bouncing motion resembling the Spring Loaded Inverted Pendulum (SLIP) model [12]. However, the stance behavior in these motions is nearly open-loop, setting only a desired leg pushoff command. This chapter introduces a different locomotion mode using stance-phase control to direct leaps and landings.

Accurate control is required to jump reliably through a complicated environment. Accurate launch velocity is required to jump to a small target, while accurate landing balance is required to remain on the target without tumbling. My collaborators and I combine high performance balance control developed in [24] and [20] with high-power jumping to achieve two behaviors: precise leaping to targets and balanced landing as shown in Fig. 4.1. [1] This is demonstrated on the jumping robot Salto-1P developed in [34] and [32].

[1] This chapter is revised from work published as "Precision robotic leaping and landing using stance-phase balance," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3422-3429, 2020 [83]. J. K. Yim was the primary author. B. R. P. Singh and R. Featherstone developed the leaning plant model and leaning control. E. K. Wang developed closed-loop leg force control used during landing and assisted in robot experiments and foot prototyping. This material is based upon work supported by Army Research Office Grant No. W911NF-18-1-0038

Related Work

This overview of related work is organized 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 [72, 13, 1, 87]. A similar approach uses steady running before launch to aim jumps [41]. 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 [44]. 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.

Significant work has demonstrated precise control of SLIP-like models in simulation [78, 23, 3] and on physical platforms [84, 73], including prior work with Salto [81]. However, the SLIP model does not capture the dynamics of balancing a landing and stopping.

Quadrupeds and hexapods with many degrees of freedom have demonstrated impressive acrobatic leaping and landing including hurdles over obstacles and flips [40, 52, 42]. 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 [17] [71] [79] 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 [10].

In walking robots, capture regions describe how to place a foot in order to arrest a robot's motion [57]. 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 [25] and [24] 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 [70, 11] and the Pendubot [68] 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 [24], the controllers in [70] and [50] employed angular momentum as the state variables for a Reaction Wheel Pendulum (RWP) in [69] and for an output zeroing controller in [50]. In [69], the controller used feedback linearization of the dynamics of a RWP and then pole-placement for the resulting linear chain of integrators, while [50] 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 [47]. In this paper, we present an angular momentum based leaning controller following from the high-performance balancing control in [24] and [20] combined with launch and landing strategies.

4.2 Models and Control Design

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. 4.2).

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. Zeroing the angular velocity limits the achievable launch angles, but partially decouples lean and leg control. In section 4.2, 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 4.2, 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 4.3.

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. 4.3). 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. Table 4.1 lists the parameters for Salto-1P. $r_{\min}$ is the minimum length of r in full

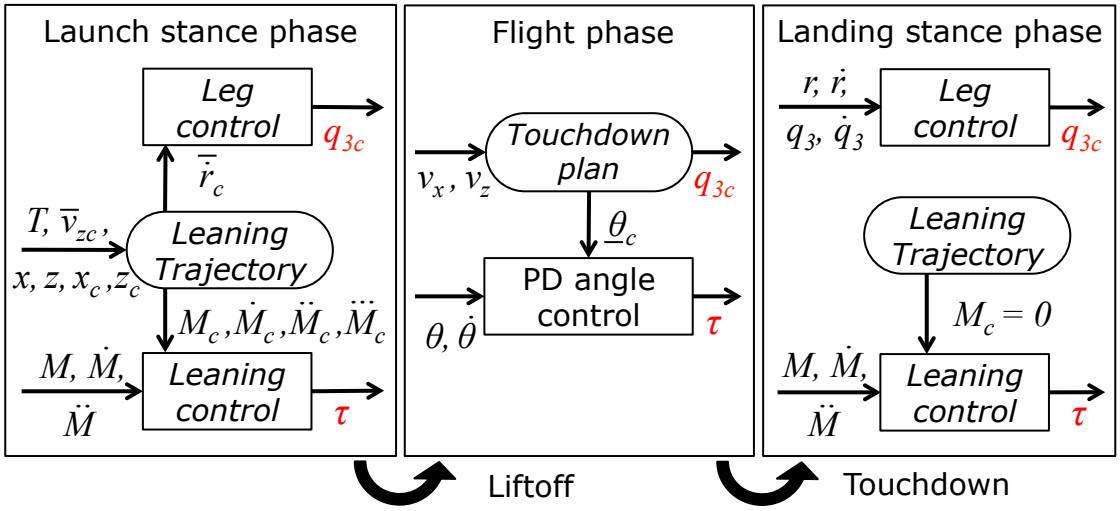


Figure 4.2: Control block diagram for launch, flight, and landing.

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. 4.3). 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.

Leaning Plant Model

For leaning, we use the plant given in [24] 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 [20] 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 [24] and [25] respectively:

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

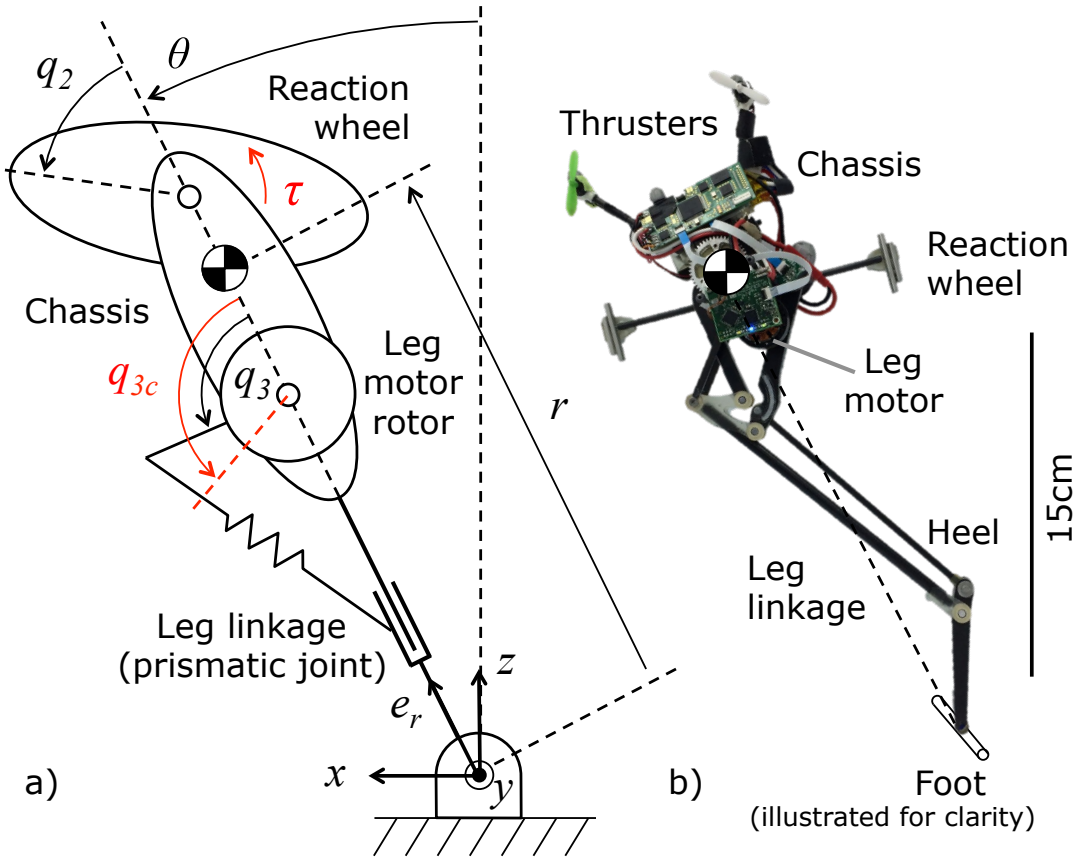


Figure 4.3: a) Planar model of the robot in stance phase. The foot behaves like a pin joint when the robot is on the ground. b) Image of Salto-1P with its components labeled.

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

$$\begin{aligned} H_{11}\ddot{\theta} + H_{12}\ddot{q}_2 &= mrg\sin(\theta) \\ H_{21}\ddot{\theta} + H_{22}\ddot{q}_2 &= \tau \end{aligned} \tag{4.1}$$

where $H_{11} = mrgT_t^2$ and $H_{12} = H_{21} = H_{22} = -G_\omega H_{11}$. Another important quantity derived in [22] 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 [22]).

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 [24] and experimentally verified on a RWP named Tippy in [20], a modified planar balance controller for the lean of the robot is presented in this subsection. This leaning controller

Table 4.1: 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 [22]	θ_{max}	rad	0.218

tracks an angular momentum command instead of the position of the actuated joint as in [24, 20]. 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:

$$M = {T_t}^2(\dot{\theta} - G_\omega \dot{q_2}), \quad \dot{M} = \theta, \quad \ddot{M} = \dot{\theta} \tag{4.2}$$

The leaning control law, similar to the controller in [24], can then be formulated based on full state feedback as:

$$\dddot{M} = k_{dd}\ddot{M} + k_d\dot{M} + k_m(M - u) \tag{4.3}$$

where u is the commanded angular momentum. The closed loop equation of motion for the control input (4.2) is:

$$\begin{bmatrix} \dddot{M} \\ \ddot{M} \\ \dot{M} \end{bmatrix} = \begin{bmatrix} k_{dd} & k_d & k_m \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \ddot{M} \\ \dot{M} \\ M \end{bmatrix} - \begin{bmatrix} k_m u \\ 0 \\ 0 \end{bmatrix} . \tag{4.4}$$

The feedback gains k_{dd}, k_d, k_m in (4.2) 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 [20]. Here the transfer function is:

$$\frac{M}{M_c} = \frac{-k_m(1 - \alpha_1 s + \alpha_2 s^2 - \alpha_3 s^3)}{(s^3 - k_{dd}s^2 - k_d s - k_m)} \tag{4.5}$$

where α_1, α_2 and α_3 are the feedforward gains. The zeros in the transfer function in (4.5) 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:

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

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 4.2, 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 (4.1):

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

As mentioned in [6, 7], the balance controllers are sensitive to the estimate of vertical, and [20] 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 (4.3) 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.

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 $\dot{\bar{r}}_c$. Thus $\bar{v}_x = \dot{\bar{r}} \sin(\bar{\theta})$ and $\bar{v}_z = \dot{\bar{r}} \cos(\bar{\theta})$. 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) $\dot{\bar{\theta}} = \dot{\bar{q}}_2 = 0$, 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 $|\theta| \leq |\theta_{\max}|$. I 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. 4.4). While many trajectories could achieve the four desired qualities, this trajectory was selected for its simple analytical solution and range of valid parameters.

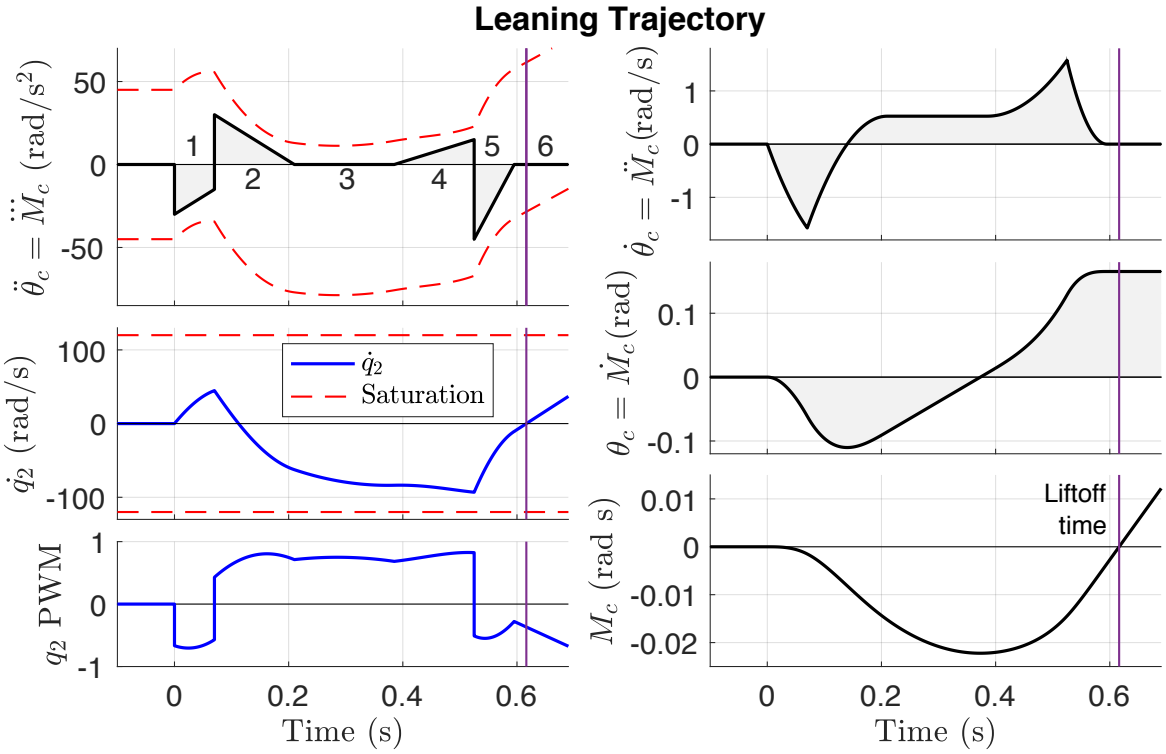
This trajectory dwells at $\bar{\theta}_c = \frac{9aT^2}{8}$ rad starting at $8.5T$ s. Liftoff should occur when $M = \dot{q}_2 = 0$, which is slightly later at $\frac{317}{36}T$ s. Due to the lean angle dwell, mistimed liftoff will perturb $\bar{L}$ and $\dot{\bar{q}}_2$ away from zero, but $\bar{\theta}$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

$$\bar{v}_{xc} = \frac{(x_c - x) g \bar{v}_{zc}}{2 r_{\max} g + \bar{v}_{zc} \sqrt{\bar{v}_{zc}^2 - 2(z_c - z) g} + \bar{v}_{zc}^2} \tag{4.6}$$

$$\bar{\theta}_c = \operatorname{atan}\left(\frac{\bar{v}_{xc}}{\bar{v}_{zc}}\right) \tag{4.7}$$

$$a = \frac{8}{9T^2}\left(\bar{\theta}_c\right) \tag{4.8}$$



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

Figure 4.4: 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.

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.

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 [25]. This relationship was found experimentally in Salto-1P and is described in section 4.3.

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 [22] after impact:

$$\underline{\theta}^{+}_{\text{eff}} = \underline{\theta} + \underline{\dot{\theta}}^{+} T_t = 0 \tag{4.9}$$

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

$$\left(m\underline{r}^2 + I_1\right)\underline{\dot{\theta}}^{+} = -m\underline{r}\left(\underline{v}_z^{-}\sin(\underline{\theta}) - \underline{v}_x^{-}\cos(\underline{\theta})\right) + I_1\underline{\dot{\theta}}^{-} \tag{4.10}$$

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

These expressions can be simplified with several assumptions: 1) We assume $\underline{\dot{\theta}}^{-}$ 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, (4.10) simplifies to:

$$\underline{\dot{\theta}}^{+} = \frac{-\underline{\theta}\underline{v}_z^{-} + \underline{v}_x^{-}}{\underline{r}} \tag{4.11}$$

Solving (4.9) and (4.11) for $\underline{\theta}$ produces the desired angle $\underline{\theta}_c$:

$$\underline{\theta}_c = -\frac{T_t\underline{v}_x^{-}}{\underline{r} - T_t\underline{v}_z^{-}} \tag{4.12}$$

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

(4.12) can also be used to find approximate bounds on $\underline{\theta}$ outside of which the robot cannot balance. Combining (4.9) and (4.11), then substituting $\pm\theta_{\max}$ for $\underline{\theta}^{+}_{\text{eff}}$ and solving for $\underline{\theta}$ yields the maximum acceptable error $\underline{\theta}_e$:

$$\underline{\theta}_e = \pm\frac{\underline{r}\theta_{\max}}{\underline{r} - T_t\underline{v}_z^-} \tag{4.13}$$

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 also contributes to touchdown angle error through (4.12). Equating (4.12) and (4.13), we solve for the maximum acceptable horizontal velocity error $\underline{v}_{\text{e}}^-$:

$$\underline{v}_{\text{e}}^- = \frac{\underline{r}\theta_{\max}}{T_t} \tag{4.14}$$

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.

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

$$\frac{\partial \bar{v}_x}{\partial \bar{\theta}} = \bar{v}_z \tag{4.15}$$

which ranged from 2.0 to 4.0 m/s per rad in Results 4.4.

For comparison, I derive a similar approximate expression for the sensitivity of flight-phase control for the Deadbeat Foot Placement Hopping Controller from Chapter 2 to touchdown angle (Fig. 4.5). Many works provide approximate solutions to the SLIP dynamics, such as [27, 63]. As in [27], 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, I approximate $\bar{\theta} - \underline{\theta} = \dot{\underline{\theta}}^{+} t_s$ with stance time t_s (0.07 s for Salto-1P). Together with (4.11) and a small angle approximation, $\bar{v}_x$ as a function of touchdown conditions becomes:

$$\bar{v}_x = \underline{v}_x^- - \underline{\theta}\underline{v}_z^- - (\underline{v}_z^- + \underline{\theta}\underline{v}_x^-)\left(\underline{\theta} + \frac{t_s(\underline{v}_x^- - \underline{\theta}\underline{v}_z^-)}{\underline{r}}\right) \tag{4.16}$$

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:

$$\frac{\partial \bar{v}_x}{\partial \underline{\theta}} = -2\underline{v}_z^- + \frac{t_s}{r}(\underline{v}_z^-)^2 \tag{4.17}$$

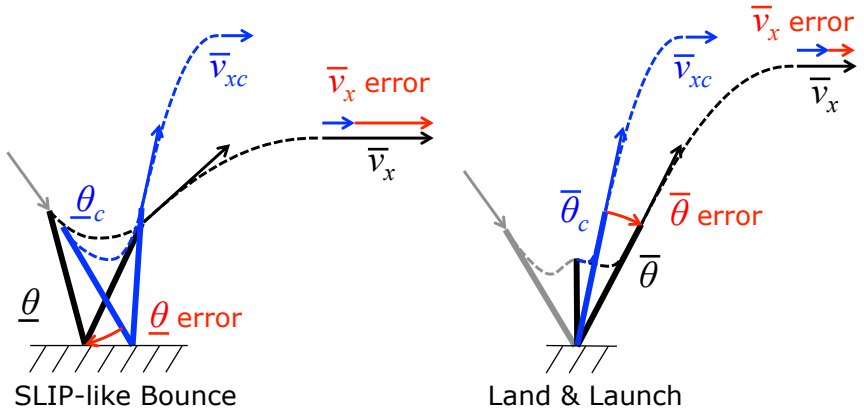


Figure 4.5: Comparison of angle error sensitivity of flight-phase control in Chapter 2 and stance-phase control in this work.

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 (4.15) is less than half the flight-phase control $\bar{v}_x$ sensitivity to $\underline{\theta}$ error in (4.17). Therefore, for small θ and similar angle errors and jump heights, stance-phase control should achieve more precise jumps than flight-phase control.

4.3 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 4.1. Salto-1P uses onboard encoders and gyroscopes to estimate q_2, q_3, r, and θ. It also uses them to estimate liftoff velocity through the SLIP Hopping Orientation and Velocity Estimator (SHOVE) as described in Chapter 3. SHOVE demonstrated a horizontal velocity error standard deviation of about 0.1 m/s, about half of the section 4.2 touchdown plan error limit. To evaluate the reliability of the leaning control, leg control, and touchdown plan with less disruption by estimator noise, motion capture augments the velocity estimation. 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.

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 [34]. 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 $\bar{\dot{r}}_c$ are related by energy:

$$\frac{1}{2}k\left(\frac{1}{G}(q_{3c}-q_o)\right)^2 = \frac{1}{2}m\bar{\dot{r}}_c^2 \tag{4.18}$$

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:

$$q_{3c} = G\sqrt{\frac{m}{k}}\bar{\dot{r}}_c + q_o \tag{4.19}$$

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

$$q_{3c} = -10v_z + 25 \tag{4.20}$$

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.

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:

$$\Delta\theta = -\text{sign}(\bar{v}_{xc})G_{\omega 3}(q_{3c} - 30) \tag{4.21}$$

$$G_{\omega 3} = -\frac{I_3}{mr_{\min}^2 + I_1 + I_2 + I_3} \approx -5 \times 10^{-4} \tag{4.22}$$

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.

4.4 Results

Launch and Landing

We tested Salto-1P's ability to launch at a grid of desired velocities and then land, shown in Fig. 4.6. Commanded launch angles ranged from -75% to +75% of $\theta_{max} = 0.218$ rad: $\bar{\theta}_c = 0, \pm 0.110$, and ± 0.165 rad ($T = 0.07$ s and $a = 0, \pm 20, \pm 30$ rad/s^2). The action of the leg motor reduced the adjusted lean angles by $5 \times 10^{-4}(q_3 - 30)$ rad for positive lean angles due to reaction wheel saturation. Commanded leg velocities $\bar{\dot{r}}_c$ 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.

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. 4.3. 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.

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 $\bar{\dot{r}}_c = 3.38$ 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. 4.7 and the resulting ballistic flight

Table 4.2: Jump Capabilities

	Previous Ch. 2	**Results 4.4**	**Results 4.4**
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

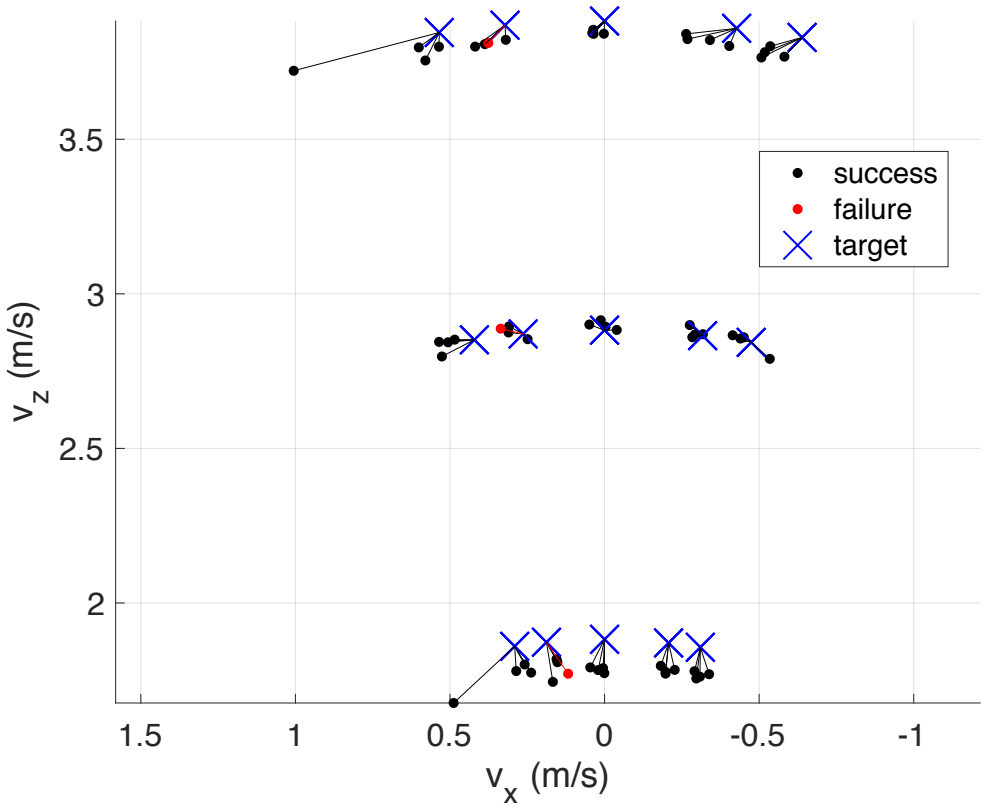


Figure 4.6: Launch and landing tests at a grid of liftoff velocities. 95% success in 60 trials (failures shown in red).

paths are shown in Fig. 4.8.

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 4.2 compares this performance to flight-phase control in Chapter 2 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 Chapter 2 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 4.2. The launch angle error for the full range of jumps in section 4.4 was worse than for the 10 moderate jumps in section 4.4 due mostly to lower accuracy at negative angles and two outliers at $a = 30$.

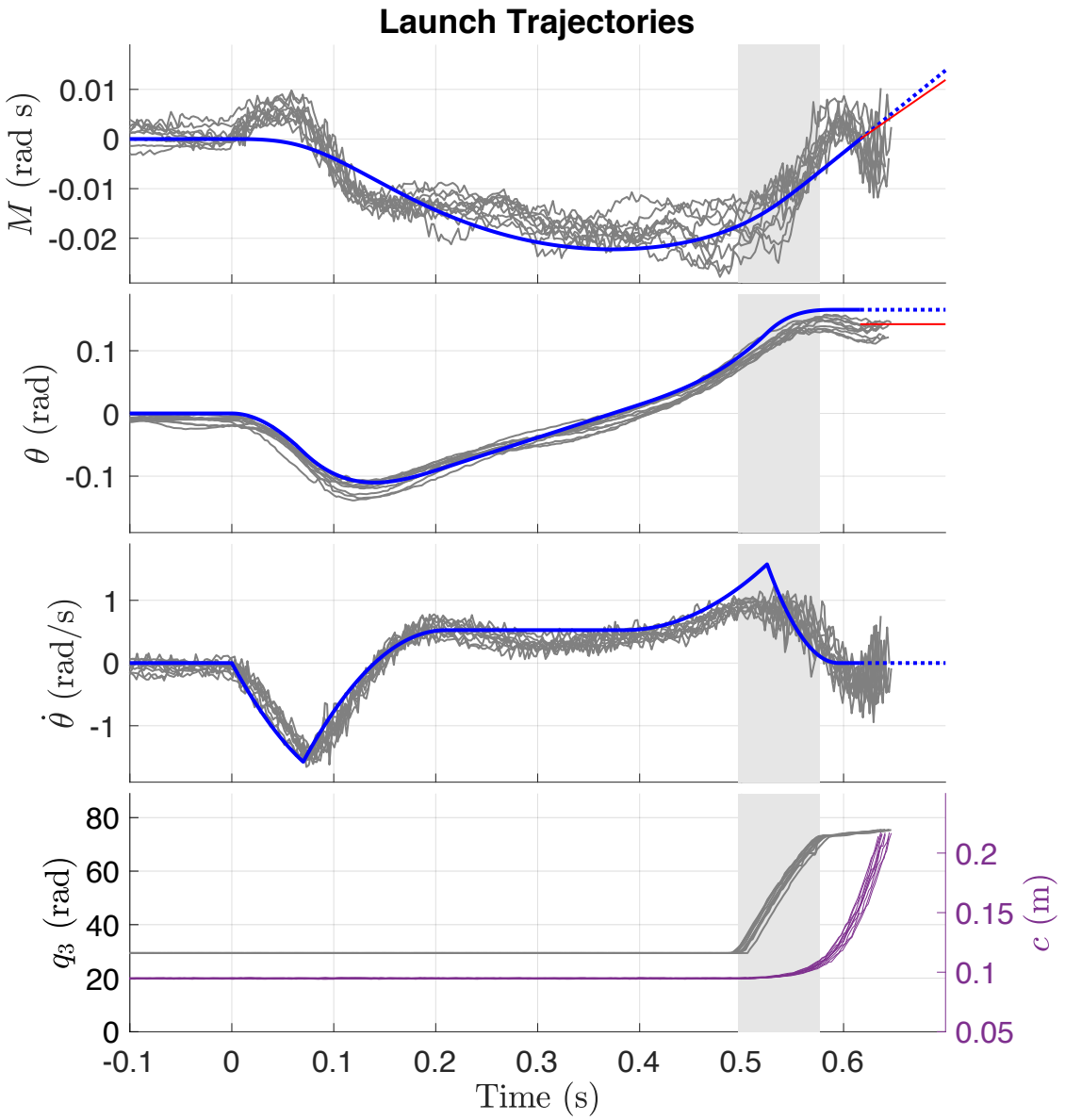


Figure 4.7: 10 launch trajectories to unadjusted launch angle 0.166 rad, adjusted angle 0.147 rad (red). Reference in blue, trajectories in grey. Duration of leg motor rotation shown as light grey.

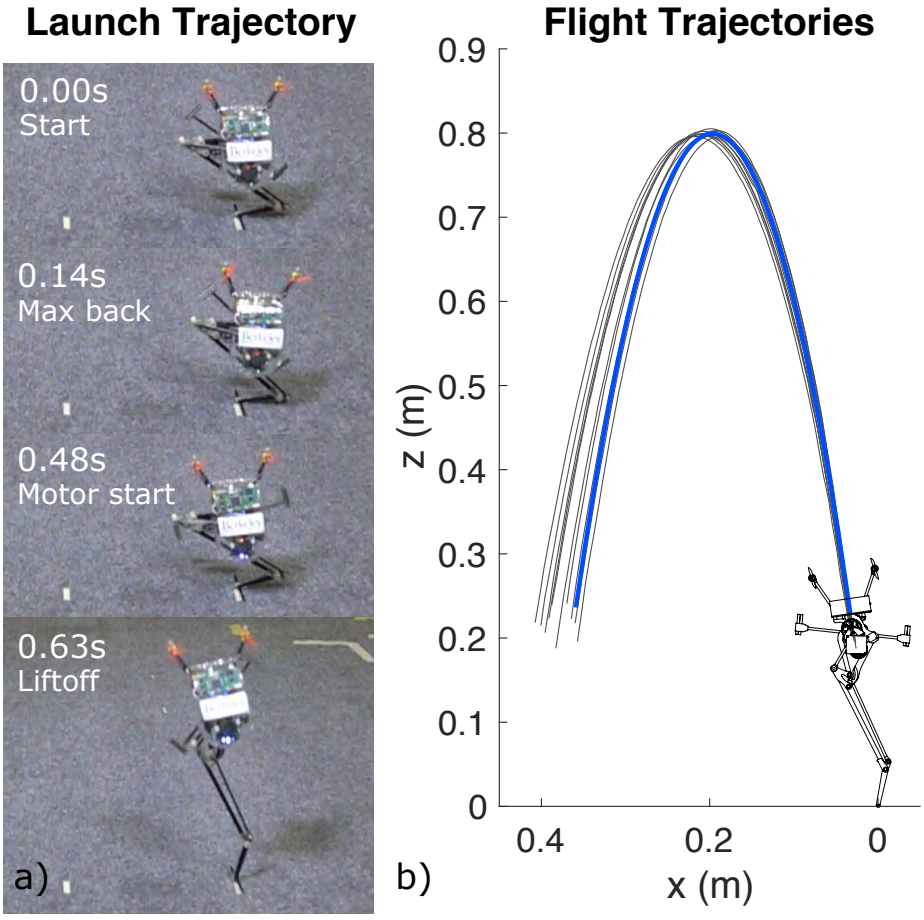


Figure 4.8: 10 jumps testing accuracy at nominal distance 32.6 cm; achieved mean 35.1 cm, standard deviation 1.6 cm. a) Launch stance trajectory. b) Resulting flight trajectories (desired in blue).

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 Chapter 2. In Fig. 4.9, 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.

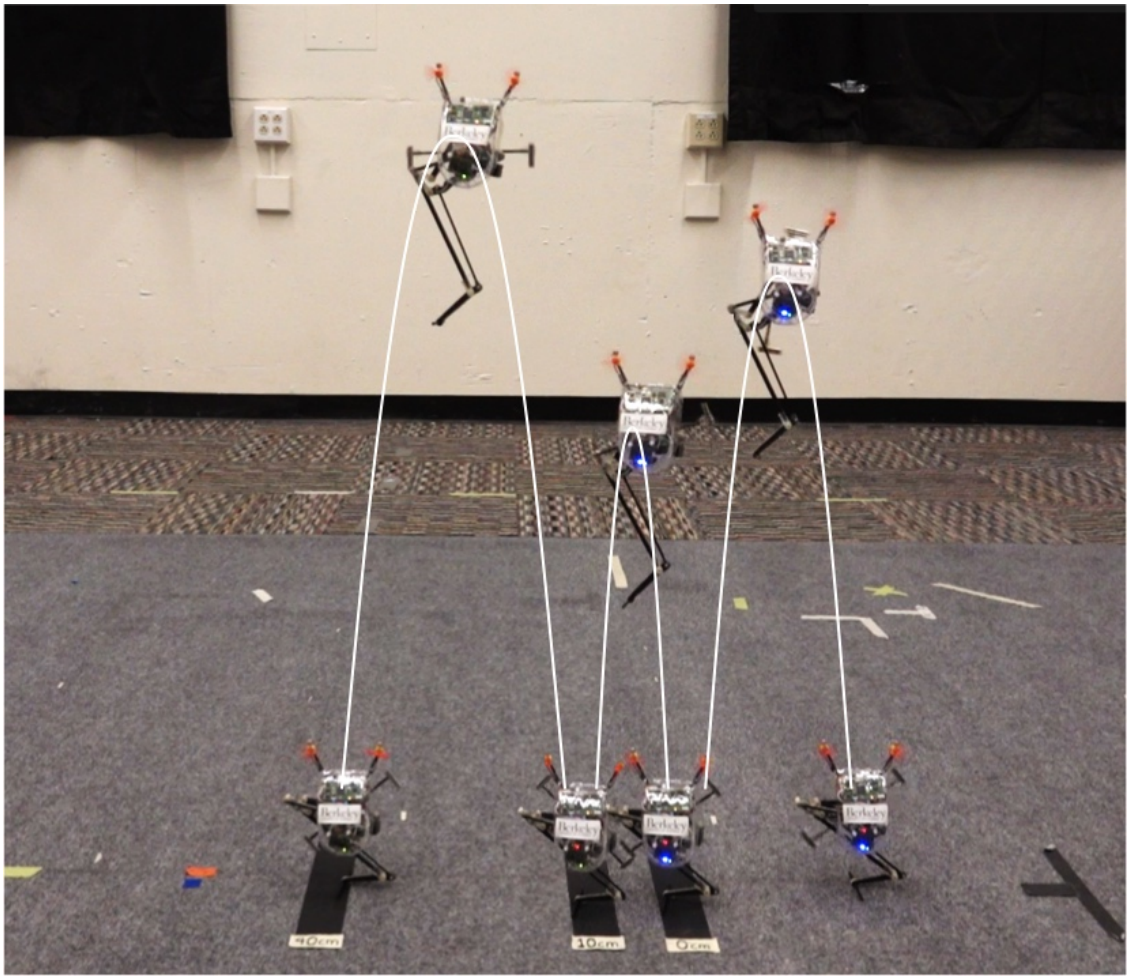


Figure 4.9: Salto-1P leaps to and lands on consecutive narrow targets marked on the floor.

4.5 Conclusion

We demonstrate accurately targeted jumping and balanced landing on a narrow support. First, this chapter demonstrates higher precision jumping to a target than that achieved in chapter 2. 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 improve the reliability of 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.

Chapter 5

Conclusions

5.1 Discussion

In this work, the monopedal hopping robot Salto-1P progressed from an initial demonstration of high-power hopping using motion capture tracking to precisely targeted hops, autonomous operation without external sensing or processing, still more precise leaps, and balanced landing transitions to standing. Experiments with Salto-1P demonstrate simple, computationally light strategies for directing and tracking high-power hopping motions that may enable small robots to move around in environments with large features.

SLIP Model

Estimation and control can be simplified for robots designed to mechanically resembled reduced-order models. Salto-1P closely resembles the Spring Loaded Inverted Pendulum (SLIP) model. In Chapter 2, SLIP-like numerical simulation transferred easily to physical operation without a large amount of tuning and adaptation required. Similarity to the SLIP model also afforded simple analysis of the relationship between leg error and running velocity in Chapter 2 that informed the SLIP Hopping Orientation and Velocity Estimator (SHOVE) attitude estimation correction in Chapter 3. Finally, the mechanical behavior of the leg, resembling a prismatic joint, enabled decoupling of leaning motions and leg extension motions in Chapter 4. Also in Chapter 4, Salto-1P's similarity to the Reaction Wheel Pendulum (RWP) allowed a simple approximate law for touchdown leg angle selection for landing and derivation of approximate bounds on acceptable velocity estimate and leg angle errors for successful landing.

In all experiments in estimation and control, leg angle error remains a salient quantity. Horizontal running velocity is highly sensitive to leg angle error (often around 0.3 m/s per degree for Salto-1P), causing foot placement inaccuracies if leg angle error is too high. However, this high sensitivity to leg angle error also means that horizontal velocity can serve as a signal by which attitude estimate error can be observed as in SHOVE. Finally, tight bounds on acceptable landing angle accuracy demonstrate why precise acrobat-like landings are so difficult to balance and highlight the impressiveness of fast motions balanced over narrow surfaces performed by squirrels and mountain goats.

By the conclusion of Chapter 4, the bulk of the control and estimation algorithms are computed by Salto-1P's onboard embedded dsPIC33FJ128MC706A microcontroller running at 40 MHz. This includes the DFPHC deadbeat velocity controller (slightly simplified), SHOVE, and the leaning control and leg touchdown control for leaping and landing. The computationally limited onboard processor is sufficient due to the simple nature of the algorithms, thanks largely to the reduced-order SLIP-like models on which they are based.

One Leg or Multiple Legs

Salto-1P's ability to execute large, precise jumps demonstrates the sufficiency of a single narrow leg for high-power hopping motion. The author conjectures that additional legs or

larger feet would not necessarily provide significant benefit for high power bouncing motion like that demonstrated by Salto. Multiple legs or a larger foot provide multiple points of contact with the ground, unlike the very SLIP-like point contact that monopeds like Salto-1P use. Usually, the benefits of multiple contact points include the ability to apply torques through force couples with the ground, redundant contacts in case one fails by sliding or dislodging, and reduced load on each individual limb. However, these benefits may not manifest for high-power bouncing hops.

Since Salto-1P spends only about 70 ms in contact with the ground per stance phase, there is very little time in which multiple contact points could apply a controlled force couple. This was the original reason behind Salto-1P's design for aerial attitude control [32].

Since Salto-1P undergoes large impacts on landing, multiple contacts with the ground could be difficult to manage and could potentially increase the complexity of touchdown conditions on complicated surfaces like rubble. While a single point of contact with the terrain means a single point of failure for sliding and stumbling, the contact dynamics are identical for any surface orientations for which ground reaction forces lie within the friction cone.

Finally, Salto-1P's single limb allows concentration of the main locomotion power source into a single actuator, possibly simplifying and streamlining the design and control.

However, multiple limbs can offer a variety of other benefits. First, additional limbs could significantly simplify the problem of getting up after a tumble; an ability Salto-1P lacks. Second, additional limbs could simplify landing. Splaying out a wide foot or three or more point feet provides a statically stable base of support that makes the challenge of balanced landing into a much simpler fully-actuated problem. Finally, additional limbs could provide additional functionality and abilities including various gaits (walking with multiple feet in contact with the ground), climbing, grasping, flying, and so on.

Scaling

Scaling is an important consideration in robot design. Greater selection and better performance in commercial off the shelf motors may be available at particular size scales [43], possibly coinciding with the scales of other commercial applications like electric vehicles or Unmanned Aerial Vehicles (UAVs). Larger robots may have a greater volume and payload capacity for sensors, processors, and other features as these components each represent a smaller fraction of the robot's space and weight for a robot with larger actuators and structures. However, as shown by Jayaram [39], small locomotors are more likely to survive uncontrolled collisions than larger locomotors due to mass, energy, and material behavior scaling. This means that a larger jumping robot would likely need to be more careful, either through more conservative motions with less chance of collision, or additional structures to minimize collision damage.

Perhaps counterintuitively, absolute jump height may change little for modest isometric scaling. Energy associated with a jump of a particular height is proportional to the robot's mass, which scales cubically with the isometric scale factor. Actuator force scales quadrati-

cally (with cross sectional area) [77] and leg stroke length scales linearly, so actuator energy also scales cubically for a jumper that is mostly limited by actuator force. Therefore, a robot's jump height is mostly unchanged by scaling according to this simple analysis. See [43, 77] and Chapter 2 of [21] for more details on actuator and jumper geometric scaling.

5.2 Future Directions

High performance hopping and its control open a variety of future behaviors and avenues for research both in the near term and long term.

Further theoretical investigation of jumping dynamics can yield more accurate understanding of jumping error sensitivity and acceptable error bounds. The DFPHC uses numerical simulations to approximate Salto-1P's dynamics, whereas SHOVE and the leaping and landing control rely on linearization. All are based on reduced-order pendulum-like models. While the simplicity of the reduced-order models used in SHOVE and the leaping and landing control make the controllers and estimators generally applicable to robots with largely similar dynamics, it reduces their accuracy for specific platforms. Frameworks like adaptive control could help refine performance as a robot continues jumping and could help reduce manual tuning and calibration, particularly as a robot ages.

Integrating precise jumping with 3D perception or Simultaneous Localization and Mapping (SLAM) and planning such as that presented by Campana and Laumond [14] can enable automated traversal of unstructured terrain. Campana and Laumond demonstrated planning for complex 3D maneuvers like chimney ascents, but no physical robot has executed such an aggressive trajectory in earth gravity. To achieve integrated planning and perception, a future jumping robot would likely need either more powerful onboard processing than Salto-1P possesses or else significant offboard processing. Many other avenues of investigation remain to produce capable jumping robot platforms including considerations in scaling, payload, and robustness.

Complex Terrains

With the ability to accurately target long jumps and balance on a small support, sparsely separated small footholds become traversable. Natural terrains of this sort include small stepping stones in a river, narrow ledges on cliffs, and branches in tree canopies. Human-constructed terrains could include areas that a robot should not touch (like delicate objects in a cluttered living space), widely separated beam-like structures like scaffolding and trusses, or disorganized rubble from collapsed buildings after a natural disaster. Jumping motion could improve a small terrestrial robot's traversal speed or enable navigation of otherwise untraversable complex terrains.

Among natural terrains, tree branches are interesting for several reasons: they often form complex canopy arrangements with varying sizes and distances, they exhibit significant compliance and may sway under excitation by a locomotor or the wind, and they are

a rich habitat to which many animals have adapted diverse locomotion methods. A better understanding of the mechanical principles of arboreal motion could shed light on the biomechanics and constraints on arboreal animals. In addition, adept robotic motion among tree branches and similar structures could be useful for pruning and agriculture, environmental monitoring and conservation, or surveillance and security. Challenges for jumping among branches include understanding of the interplay between variable (possibly anisotropic and nonlinear) branch compliance and jumper dynamics, improved jump targeting, grasping and perching limbs or end effectors, perception and planning to navigate canopies, and strategies for recovery in the event of an off-target jump.

Agile legged robots

Small, capable legged robots that can navigate their environments with agile motions like jumping can open new opportunities for robotics and related research.

Small legged robots offer different advantages than similarly-sized Micro Aerial Vehicles (MAVs) that might be employed for similar tasks. Aerial vehicles have several advantages of their own: they have supreme freedom of movement since they do not require surfaces on which to apply forces, many can easily move much faster than similarly-sized legged robots, and—at least among animals—flying often achieves a much lower cost of transport than walking or running [74], though this may or may not hold among robots. However, legged robots may be preferable in situations where MAVs could be disruptive or in those that require close or forceful interactions with the environment.

Lighter than air MAVs must have a large volume relative to their payload to generate buoyant force. Consequently, they may not fit in confined areas where humans might want to send robots, or be too bulky in some environments alongside humans. Heavier than air MAVs must necessarily generate downwash as a byproduct of the lift that supports them. Particularly substantial downwash, like that beneath a rotorcraft in hover, could disrupt light objects like papers in an office, kick up dust and debris in natural environments, or fan embers and flames in disaster scenarios. Finally, MAVs may simply be unwanted in some everyday scenarios due to their flight paths or noise.

Legged robots regularly experience large, impulse-like forces many times greater than their bodyweight during locomotion and can apply large, controlled forces with their appendages. This may make them better suited to close interaction with cluttered terrain like dense undergrowth in which unexpected contacts are frequent and forceful manipulation of the environment may be required for forward progress. In addition, agile legged robots' powerful actuators and appendages may be repurposed for forceful interactions like pushing, pulling, breaking, or digging.

Besides direct utility in applications, legged robots can also provide an avenue for insight into biological systems. Some hypotheses may be difficult to test directly in an animal due to the interrelated and multifunctional nature of many systems: a limb may be used for walking, feeding, preening, mating, and more, complicating disambiguation of different requirements and pressures. Some parameters of an animal may be difficult to modify

and some measurements may be difficult to obtain without invasive operations that could significantly alter an animal's behavior. An analogous robot capable of reproducing relevant aspects of an animal's performance can serve as a convenient instrumented test subject that can be modified as required for some experiments that would be difficult to set up in a biological system [38].

Creating robots that can perform like natural systems is one of the ten grand challenges identified by AAAS Science Robotics in 2018 [80]. This dissertation develops control and estimation for high-power jumping motion using reduced-order models with the goal of achieving animal-like nimbleness. Ongoing improvement in the understanding of the mechanics and control of agile legged locomotion can enable robots to accompany humans and animals everywhere they go, and beyond.

Bibliography

[1] E Ackerman. *Boston Dynamics sand flea robot demonstrates astonishing jumping skills*. 2012. URL: https://spectrum.ieee.org/automaton/robotics/military-robots/boston-dynamics-sand-flea-demonstrates-astonishing-jumping-skills.

[2] Peter Aerts. "Vertical jumping in Galago senegalensis: the quest for an obligate mechanical power amplifier". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 353.1375 (1998), pp. 1607–1620. ISSN: 0962-8436. DOI: 10.1098/rstb.1998.0313. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rstb.1998.0313.

[3] Ayush Agrawal and Koushil Sreenath. "Bipedal Robotic Running on Stochastic Discrete Terrain". In: *European Control Conference (ECC)*. IEEE, 2019, pp. 3564–3570. ISBN: 978-3-907144-00-8. DOI: 10.23919/ECC.2019.8795938. URL: https://ieeexplore.ieee.org/document/8795938/.

[4] M. Ahmadi and M. Buehler. "The ARL monopod II running robot: control and energetics". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 3. IEEE, 1999, pp. 1689–1694. ISBN: 0-7803-5180-0. DOI: 10.1109/ROBOT.1999.770352. URL: http://ieeexplore.ieee.org/document/770352/.

[5] Ömür Arslan and Uluç Saranli. "Reactive Planning and Control of Planar Spring-Mass Running on Rough Terrain". In: *IEEE Transactions on Robotics* 28.3 (2012), pp. 567–579. ISSN: 1552-3098. DOI: 10.1109/TRO.2011.2178134. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6112244.

[6] Morteza Azad. "Balancing and Hopping Motion Control Algorithms for an Underactuated Robot". PhD thesis. Australian National University, 2014.

[7] Morteza Azad and Roy Featherstone. "Angular momentum based balance controller for an under-actuated planar robot". In: *Autonomous Robots* 40.1 (2016), pp. 93–107. ISSN: 15737527. DOI: 10.1007/s10514-015-9446-z.

[8] A.-J. Baerveldt and R Klang. "A Low-cost and Low-weight Attitude Estimation System for an Autonomous Helicopter". In: *IEEE International Conference on Intelligent Engineering Systems*. 1997, pp. 391–395. ISBN: 0780336275. DOI: 10.1109/INES.1997.632450.

[9] Zachary Batts, Joohyung Kim, and Katsu Yamane. "Untethered One-Legged Hopping in 3D Using Linear Elastic Actuator in Parallel (LEAP)". In: *International Symposium on Experimental Robotics*. 2017, pp. 103–112. DOI: 10.1007/978-3-319-50115-4_10. URL: http://link.springer.com/10.1007/978-3-319-50115-4{_}10.

[10] M.D. Berkemeier and R.S. Fearing. "Sliding and hopping gaits for the underactuated Acrobot". In: *IEEE Transactions on Robotics and Automation* 14.4 (1998), pp. 629–634. ISSN: 1042296X. DOI: 10.1109/70.704235. URL: http://ieeexplore.ieee.org/document/704235/.

[11] M.D. Berkemeier and R.S. Fearing. "Tracking fast inverted trajectories of the underactuated Acrobot". In: *IEEE Transactions on Robotics and Automation* 15.4 (1999), pp. 740–750. ISSN: 1042296X. DOI: 10.1109/70.782028. URL: http://ieeexplore.ieee.org/document/782028/.

[12] Reinhard Blickhan. "the Spring-Mass Model for Running and Hopping". In: *Journal of Biomechanics* 22.1112 (1989), pp. 1217–1227.

[13] Joel Burdick and Paolo Fiorini. "Minimalist jumping robots for celestial exploration". In: *International Journal of Robotics Research* 22.7-8 (2003), pp. 653–674. ISSN: 0278-3649. DOI: 10.1177/02783649030227013.

[14] Mylene Campana and Jean-paul Laumond. "Ballistic motion planning". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, pp. 1410–1416. ISBN: 9781509037612. DOI: 10.1109/IROS.2016.7759230.

[15] Sean G. Carver, Noah J. Cowan, and John M. Guckenheimer. "Lateral stability of the spring-mass hopper suggests a two-step control strategy for running". In: *Chaos* 19.2 (2009). ISSN: 10541500. DOI: 10.1063/1.3127577.

[16] George Council, Shiyi Yang, and Shai Revzen. "Deadbeat control with (almost) no sensing in a hybrid model of legged locomotion". In: *International Conference on Advanced Mechatronic Systems, ICAMechS*. 2014, pp. 475–480. ISBN: 9781479963812. DOI: 10.1109/ICAMechS.2014.6911592.

[17] Hongkai Dai, Andrés Valenzuela, and Russ Tedrake. "Whole-body motion planning with centroidal dynamics and full kinematics". In: *IEEE-RAS International Conference on Humanoid Robots*. Vol. 2015-Febru. 2015, pp. 295–302. ISBN: 9781479971749. DOI: 10.1109/HUMANOIDS.2014.7041375.

[18] Avik De and Daniel E. Koditschek. "Parallel composition of templates for tail-energized planar hopping". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 4562–4569. ISBN: VO -. DOI: 10.1109/ICRA.2015.7139831.

[19] Amir Degani et al. "The ParkourBot – A Dynamic BowLeg Climbing Robot". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2011. ISBN: 9781612843858. DOI: 10.1109/ICRA.2011.5979937.

[20] Josephus J. M. Driessen et al. "Experimental Demonstration of High-Performance Robotic Balancing". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9459–9465. ISBN: 978-1-5386-6027-0. DOI: 10.1109/ICRA.2019.8794447. URL: https://ieeexplore.ieee.org/document/8794447/.

[21] Josephus J.M. Driessen. "Design of high-performance legged robots". PhD thesis. University of Genova and Istituto Italiano di Tecnologia, 2019.

[22] Josephus J.M. Driessen, Roy Featherstone, and Antonios 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. ISBN: 9781538680940. DOI: 10.1109/IROS.2018.8593729.

[23] Johannes Englsberger et al. "Biologically Inspired Deadbeat Control for Running: From Human Analysis to Humanoid Control and Back". In: *IEEE Transactions on Robotics* 32.4 (2016), pp. 854–867. ISSN: 1552-3098. DOI: 10.1109/TRO.2016.2581199. URL: http://ieeexplore.ieee.org/document/7527657/.

[24] Roy Featherstone. "A simple model of balancing in the plane and a simple preview balance controller". In: *International Journal of Robotics Research* 36.13-14 (2017), pp. 1489–1507. ISSN: 17413176. DOI: 10.1177/0278364917691114.

[25] Roy Featherstone. "Quantitative measures of a robot's physical ability to balance". In: *International Journal of Robotics Research* 35.14 (2016), pp. 1681–1696. ISSN: 17413176. DOI: 10.1177/0278364916669599.

[26] Eric Foxlin. "Pedestrian tracking with shoe-mounted inertial sensors". In: *IEEE Computer Graphics and Applications* 25.6 (2005), pp. 38–46. ISSN: 02721716. DOI: 10.1109/MCG.2005.140.

[27] Hartmut Geyer, Andre Seyfarth, and Reinhard Blickhan. "Spring-mass running: Simple approximate solution and application to gait stability". In: *Journal of Theoretical Biology* 232.3 (2005), pp. 315–328. ISSN: 00225193. DOI: 10.1016/j.jtbi.2004.08.015.

[28] Pedro Gregorio, Mojtaba Ahmadi, and Martin Buehler. "Design, control, and energetics of an electrically actuated legged robot". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 27.4 (1997), pp. 626–634. ISSN: 10834419. DOI: 10.1109/3477.604106.

[29] Jesse A. Grimes and Jonathan W. Hurst. "The Design of ATRIAS 1.0 a Unique Monopod, Hopping Robot". In: *Adaptive Mobile Robotics*. World Scientific, 2012, pp. 548–554. ISBN: 9789814415941. DOI: 10.1142/9789814415958_0071. URL: http://www.worldscientific.com/doi/abs/10.1142/9789814415958{_}0071.

[30] J.W. Grizzle et al. "MABEL, a new robotic bipedal walker and runner". In: *American Control Conference*. IEEE, 2009, pp. 2030–2036. ISBN: 978-1-4244-4523-3. DOI: 10.1109/ACC.2009.5160550. URL: http://ieeexplore.ieee.org/document/5160550/.

[31] Radha Charan Gupta. "Bhaskara I's Approximation to Sine". In: *Indian Journal of History of Science* 2.2 (1967), pp. 121–136.

[32] Duncan W Haldane, Justin K Yim, and Ronald S Fearing. "Repetitive extreme-acceleration (14-g) spatial jumping with Salto-1P". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 3345–3351. ISBN: 978-1-5386-2682-5. DOI: 10.1109/IROS.2017.8206172. URL: http://ieeexplore.ieee.org/document/8206172/.

[33] Duncan W Haldane et al. "A power modulating leg mechanism for monopedal hopping". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, pp. 4757–4764. ISBN: 9781509037629. DOI: 10.1109/IROS.2016.7759699.

[34] Duncan W Haldane et al. "Robotic vertical jumping agility via series-elastic power modulation". In: *Science Robotics* 1.1 (2016), eaag2048. ISSN: 2470-9476. DOI: 10.1126/scirobotics.aag2048. URL: http://robotics.sciencemag.org/content/robotics/1/1/eaag2048.full.pdf.

[35] Jessica K. Hodgins and Marc H Raibert. "Adjusting Step Length for Rough Terrain Locomotion". In: *IEEE Transactions on Robotics and Automation* 7.3 (1991), pp. 289–298. ISSN: 1042296X. DOI: 10.1109/70.88138.

[36] Christian Hubicki et al. "ATRIAS: Design and validation of a tether-free 3D-capable spring-mass bipedal robot". In: *International Journal of Robotics Research* 35.12 (2016), pp. 1497–1521. ISSN: 0278-3649. DOI: 10.1177/0278364916648388. URL: http://journals.sagepub.com/doi/10.1177/0278364916648388.

[37] S.H. Hyon and T. Mita. "Development of a biologically inspired hopping robot-"Kenken"". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 4. IEEE, 2002, pp. 3984–3991. ISBN: 0-7803-7272-7. DOI: 10.1109/ROBOT.2002.1014356. URL: http://ieeexplore.ieee.org/document/1014356/.

[38] Auke J. Ijspeert. "Biorobotics: Using robots to emulate and investigate agile locomotion". In: *Science* 346.6206 (2014), pp. 196–203. ISSN: 0036-8075. DOI: 10.1126/science.1254486. URL: https://www.sciencemag.org/lookup/doi/10.1126/science.1254486.

[39] Kaushik Jayaram et al. "Transition by head-on collision: mechanically mediated manoeuvres in cockroaches and small robots". In: *Journal of the Royal Society Interface* 15.139 (2018). ISSN: 17425662. DOI: 10.1098/rsif.2017.0664.

[40] Aaron M Johnson and Daniel E. Koditschek. "Toward a vocabulary of legged leaping". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2013, pp. 2568–2575. ISBN: 9781467356411. DOI: 10.1109/ICRA.2013.6630928.

[41] Gwang-Pil Jung et al. "An integrated jumping-crawling robot using height-adjustable jumping module". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4680–4685. ISBN: 978-1-4673-8026-3. DOI: 10.1109/ICRA.2016.7487668. URL: http://ieeexplore.ieee.org/document/7487668/.

[42] Benjamin Katz, Jared Di Carlo, and Sangbae Kim. "Mini Cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6295–6301. ISBN: 978-1-5386-6027-0. DOI: 10.1109/ICRA.2019.8793865. URL: https://ieeexplore.ieee.org/document/8793865/.

[43] Gavin Kenneally, Avik De, and D. E. Koditschek. "Design Principles for a Family of Direct-Drive Legged Robots". In: *IEEE Robotics and Automation Letters* 1.2 (2016), pp. 900–907. ISSN: 2377-3766. DOI: 10.1109/LRA.2016.2528294. URL: http://ieeexplore.ieee.org/document/7403902/.

[44] Mirko Kovač et al. "The EPFL jumpglider: A hybrid jumping and gliding robot with rigid or folding wings". In: *IEEE International Conference on Robotics and Biomimetics, ROBIO 2011*. IEEE, 2011, pp. 1503–1508. ISBN: 9781457721373. DOI: 10.1109/ROBIO.2011.6181502.

[45] Dominic Lakatos et al. "Targeted jumping of compliantly actuated hoppers based on discrete planning and switching control". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 5802–5808. ISBN: 978-1-4799-9994-1. DOI: 10.1109/IROS.2015.7354201. URL: http://ieeexplore.ieee.org/document/7354201/.

[46] E.J. Lefferts, F.L. Markley, and M.D. Shuster. "Kalman Filtering for Spacecraft Attitude Estimation". In: *Journal of Guidance, Control, and Dynamics* 5.5 (1982), pp. 417–429. ISSN: 0731-5090. DOI: 10.2514/3.56190. URL: http://arc.aiaa.org/doi/10.2514/3.56190.

[47] Macchietto and Adriano Patrick. "Momentum-Based Balance Control For Simulated Characters". Master of Science. University of California, Riverside, 2008.

[48] Hector Garcia de Marina et al. "UAV Attitude Estimation Using Unscented Kalman Filter and TRIAD". In: *IEEE Transactions on Industrial Electronics* 59.11 (2012), pp. 4465–4474. ISSN: 0278-0046. DOI: 10.1109/TIE.2011.2163913. arXiv: 1609.07436. URL: http://ieeexplore.ieee.org/document/5977026/.

[49] Thomas A. McMahon and George C. Cheng. "The mechanics of running: How does stiffness couple with speed?" In: *Journal of Biomechanics* 23.Supplement 1 (1990), pp. 65–78. ISSN: 00219290. DOI: 10.1016/0021-9290(90)90042-2. URL: https://linkinghub.elsevier.com/retrieve/pii/0021929090900422.

[50] Naoko Miyashita, Masashi Kishikawa, and Masaki Yamakita. "3D motion control of 2 links (5 D.O.F.) underactuated manipulator named AcroBOX". In: *American Control Conference*. IEEE, 2006, pp. 5614–5619. ISBN: 1424402107. DOI: 10.1109/ACC.2006.1657618.

[51] J. Gordon Nichol et al. "System design of a quadrupedal galloping machine". In: *International Journal of Robotics Research* 23.10-11 (2004), pp. 1013–1027. ISSN: 02783649. DOI: 10.1177/0278364904047391.

[52] Hae Won Park, Patrick M. Wensing, and Sangbae Kim. "Online planning for autonomous running jumps over obstacles in high-speed quadrupeds". In: *Robotics: Science and Systems* 11 (2015). ISSN: 2330765X. DOI: 10.15607/RSS.2015.XI.047.

[53] Rolf Pfeifer, Max Lungarella, and Fumiya Iida. "Self-Organization, Embodiment, and Biologically Inspired Robotics". In: *Science* 318.5853 (2007), pp. 1088–1093. ISSN: 0036-8075. DOI: 10.1126/science.1145803. URL: https://www.sciencemag.org/lookup/doi/10.1126/science.1145803.

[54] Giulia Piovan and Katie Byl. "Reachability-based control for the active SLIP model". In: *International Journal of Robotics Research* 34.3 (2015), pp. 270–287. ISSN: 0278-3649. DOI: 10.1177/0278364914552112. URL: http://journals.sagepub.com/doi/10.1177/0278364914552112.

[55] Mark M. Plecnik et al. "Design exploration and kinematic tuning of a power modulating jumping monopod". In: *Journal of Mechanisms and Robotics* 9.1 (2016), pp. 1–13. ISSN: 1942-4302. DOI: 10.1115/1.4035117.

[56] Ioannis Poulakakis and J. W. Grizzle. "Modeling and control of the monopedal robot Thumper". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 3327–3334. ISBN: 9781424427895. DOI: 10.1109/robot.2009.5152708.

[57] Jerry Pratt et al. "Capture point: A step toward humanoid push recovery". In: *IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2006, pp. 200–207. ISBN: 142440200X. DOI: 10.1109/ICHR.2006.321385.

[58] Marc H Raibert and Jr. H. B. Brown. "Experiments in Balance With a 3D One-Legged Hopping Machine". In: *Journal of Dynamic Systems, Measurement, and Control* 106.1 (1984), pp. 75–81. ISSN: 0022-0434. DOI: 10.1115/1.3149668. URL: http://dx.doi.org/10.1115/1.3149668.

[59] Marc H Raibert and Francis C Wimberly. "Tabular Control of Balance in a Dynamic Legged System". In: *IEEE Transactions on Systems, Man, and Cybernetics* 2 (1984), pp. 334–339.

[60] Henrik Rehbinder and Xiaoming Hu. "Drift-free attitude estimation for accelerated rigid bodies". In: *Automatica* 40.4 (2004), pp. 653–659. ISSN: 00051098. DOI: 10.1016/j.automatica.2003.11.002.

[61] David Rollinson et al. "Design and Modeling of a Series Elastic Element for Snake Robots". In: *Proceedings of the ASME 2013 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2013. ISBN: 978-0-7918-5612-3. DOI: 10.1115/DSCC2013-3875. URL: http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?doi=10.1115/DSCC2013-3875.

[62] Uluc Saranli, William J. Schwind, and Daniel E. Koditschek. "Toward the control of a multi-jointed, monoped runner". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 1998, pp. 2676–2682. ISBN: 0-7803-4300-X. DOI: 10.1109/ROBOT.1998.680750. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=680750.

[63] Uluç Saranli et al. "Approximate analytic solutions to non-symmetric stance trajectories of the passive Spring-Loaded Inverted Pendulum with damping". In: *Nonlinear Dynamics* 62.4 (2010), pp. 729–742. ISSN: 0924090X. DOI: 10.1007/s11071-010-9757-8.

[64] Ajij Sayyad, B. Seth, and P. Seshu. "Single-legged hopping robotics research—A review". In: *Robotica* 25.05 (2007), pp. 587–613. ISSN: 0263-5747. DOI: 10.1017/S0263574707003487. URL: http://www.journals.cambridge.org/abstract{_}S0263574707003487.

[65] W. J. Schwind and Daniel E. Koditschek. "Approximating the stance map of a 2-DOF monoped runner". In: *Journal of Nonlinear Science* 10.5 (2000), pp. 533–568. ISSN: 09388974. DOI: 10.1007/s004530010001.

[66] André Seyfarth, Hartmut Geyer, and Hugh Herr. "Swing-leg retraction: a simple control model for stable running". In: *Journal of Experimental Biology* 206.15 (2003), pp. 2547–2555. ISSN: 0022-0949. DOI: 10.1242/jeb.00463. URL: http://jeb.biologists.org/cgi/doi/10.1242/jeb.00463.

[67] Natan Shemer and Amir Degani. "A flight-phase terrain following control strategy for stable and robust hopping of a one-legged robot under large terrain variations". In: *Bioinspiration and Biomimetics* 12.4 (2017), aa741f. ISSN: 17483190. DOI: 10.1088/1748-3190/aa741f. URL: https://doi.org/10.1088/1748-3190/aa741f.

[68] Mark W Spong and Daniel J Block. "The Pendubot: A Mechatronic System". In: *Proceedings of the 34th Conference on Decision & Control New Orleans, LA*. 1995, pp. 555–556. ISBN: 0780326857.

[69] Mark W. Spong, Peter Corke, and Rogelio Lozano. "Nonlinear control of the Reaction Wheel Pendulum". In: *Automatica* 37.11 (2001), pp. 1845–1851. ISSN: 00051098. DOI: 10.1016/S0005-1098(01)00145-5.

[70] M.W. Spong. "Swing up control of the Acrobot". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 1994, pp. 2356–2361. ISBN: 0-8186-5330-2. DOI: 10.1109/ROBOT.1994.350934. URL: http://ieeexplore.ieee.org/document/350934/.

[71] Koushil Sreenath et al. "Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on MABEL". In: *International Journal of Robotics Research* 32.3 (2013), pp. 324–345. ISSN: 02783649. DOI: 10.1177/0278364912473344.

[72] Sascha A Stoeter et al. "Autonomous stair-hopping with scout robots". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2002, pp. 721–726. ISBN: 0780373987. DOI: 10.1109/IRDS.2002.1041476. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1041476.

[73] Pat Terry, Giulia Piovan, and Katie 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*. IEEE, 2016, pp. 6669–6675. ISBN: 9781509018376. DOI: 10.1109/CDC.2016.7799296.

[74] V. A. Tucker. "The Energetic Cost of Moving About: Walking and running are extremely inefficient forms of locomotion. Much greater efficiency is achieved by birds, fish—and bicyclists". In: *American Scientist* 63.4 (1975), pp. 413–419. ISSN: 00030996.

[75] Ismail Uyanik, Ömer Morgül, and Uluc Saranli. "Experimental validation of a feedforward predictor for the spring-loaded inverted pendulum template". In: *IEEE Transactions on Robotics* 31.1 (2015), pp. 208–216. ISSN: 15523098. DOI: 10.1109/TRO.2014.2383531.

[76] Jonathan Van Why. "IMU Integration for ATRIAS". PhD thesis. Oregon State University, 2016.

[77] Patrick M. Wensing et al. "Proprioceptive Actuator Design in the MIT Cheetah: Impact Mitigation and High-Bandwidth Physical Interaction for Dynamic Legged Robots". In: *IEEE Transactions on Robotics* 33.3 (2017), pp. 509–522. ISSN: 1552-3098. DOI: 10.1109/TRO.2016.2640183. URL: http://ieeexplore.ieee.org/document/7827048/.

[78] Albert Wu and Hartmut Geyer. "The 3-D spring-mass model reveals a time-based deadbeat control for highly robust running and steering in uncertain environments". In: *IEEE Transactions on Robotics* 29.5 (2013), pp. 1114–1124. ISSN: 15523098. DOI: 10.1109/TRO.2013.2263718.

[79] Xiaobin Xiong and Aaron D. Ames. "Bipedal Hopping: Reduced-Order Model Embedding via Optimization-Based Control". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 3821–3828. ISBN: 9781538680940. DOI: 10.1109/IROS.2018.8593547. arXiv: 1807.08037.

[80] Guang-Zhong Yang et al. "The grand challenges of Science Robotics". In: *Science Robotics* 3.14 (2018), eaar7650. ISSN: 2470-9476. DOI: 10.1126/scirobotics.aar7650. URL: https://robotics.sciencemag.org/lookup/doi/10.1126/scirobotics.aar7650.

[81] Justin K Yim and Ronald 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. ISBN: 978-1-5386-8094-0. DOI: 10.1109/IROS.2018.8594154. URL: https://ieeexplore.ieee.org/document/8594154/.

[82] Justin K Yim, Eric K Wang, and Ronald S Fearing. "Drift-free Roll and Pitch Estimation for High-acceleration Hopping". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8986–8992. ISBN: 978-1-5386-6027-0. DOI: 10.1109/ICRA.2019.8793259. URL: https://people.eecs.berkeley.edu/{~}ronf/PAPERS/jyim-icra2019.pdf.

[83] Justin K. Yim et al. "Precision robotic leaping and landing using stance-phase balance". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3422–3429. ISSN: 23773766. DOI: 10.1109/LRA.2020.2976597.

[84] Garth Zeglin. "The Bow Leg Hopping Robot". PhD thesis. Carnegie Mellon University, 1999.

[85] Garth Zeglin. "Uniroo: A One Legged Dynamic Hopping Robot". Bachelor's Thesis. Massachusetts Institute of Technology, 1991.

[86] Garth Zeglin and H Benjamin Brown. "First Hops of the 3D Bow Leg Hopper". In: *The 5th International Conference on Climbing and Walking Robots* (2002), pp. 357–364.

[87] Jianguo Zhao et al. "MSU jumper: A single-motor-actuated miniature steerable jumping robot". In: *IEEE Transactions on Robotics* 29.3 (2013), pp. 602–614. ISSN: 15523098. DOI: 10.1109/TRO.2013.2249371.