Drift-free Roll and Pitch Estimation for High-acceleration Hopping

Justin K. Yim[1], Eric K. Wang[2], and Ronald S. Fearing[1]

***Abstract*— We develop a drift-free roll and pitch attitude estimation scheme for monopedal jumping robots. The estimator uses only onboard rate gyroscopes and encoders and does not rely on external sensing or processing. It is capable of recovering from attitude estimate disturbances and, together with onboard velocity estimation, enables fully autonomous stable hopping control. The estimator performs well on a small untethered robot capable of large jumps and extreme stance accelerations. We demonstrate that the robot can follow a rectangular path using onboard dead-reckoning with less than 2 meters of drift over 200 seconds and 300 jumps covering 60 m. We also demonstrate that the robot can operate untethered outdoors under human wireless joystick direction.**

I. Introduction

A. Motivation

Small legged robots offer compelling potential for mobility in challenging environments. Legged robots can traverse terrain impassable to other ground vehicles like those using wheels or tracks. Smaller robots can be cheaper and less intrusive, pass through smaller apertures, and are more robust to falls and collisions [9]. By jumping, legged robots can overcome obstacles or gaps longer than their maximum dimensions, allowing small robots to maneuver in complex human-scale or natural environments.

Animal locomotors are adept at continuous hopping to navigate difficult environments like trees or cliffs. However, current small robots cannot match the feats of animals. Untethered electric robots fall short of animals' vertical jumping agility [7]. 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 [12] [19] [15] [2].

In this work we develop an onboard SLIP Hopping Orientation and Velocity Estimator (SHOVE) for a 0.10 kg, 0.31 m monopedal jumping robot to achieve stable hopping without the support of external sensing or processing.

B. Prior Work

Significant prior work has investigated the dynamics and control of running and hopping. We draw on the Spring Loaded Inverted Pendulum (SLIP) model described in [3]. We also adopt the framework of Raibert's controller by selecting touchdown leg angles to direct a SLIP-like hopping robot [12]. This flight-phase control is effective even for jumping robots with short impulse-like stance phases [8].

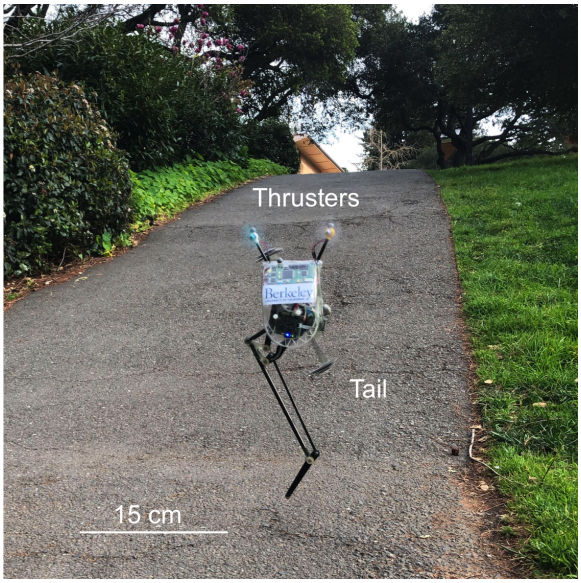

Fig. 1: Salto-1P jumps outdoors without external sensing.

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 [2]. 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 [4] and complementary filters [1] 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 [10]. These sensors require mostly unobstructed views to their known references and would not be appropriate for a terrestrial robot in a cluttered environment.

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. [13]. Walking humans have been similarly tracked with shoe-mounted inertial sensors [5].

Running robots sometimes use other sensors to augment rate gyroscopes like an infrared sensor pointed at the ground [11]. Others rely on low gyroscope bias so that the drift is small over their operating duration [16].

This material is based upon work supported by Army Research Office Grant No. W911NF-18-1-0038.

[1]J.K. Yim, and R.S. Fearing are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA yim@eecs.berkeley.edu, ronf@eecs.berkeley.edu

[2]E.K. Wang is with the Department of Mechanical Engineering, University of California, Berkeley, CA 94720 USA eric.wang21@berkeley.edu

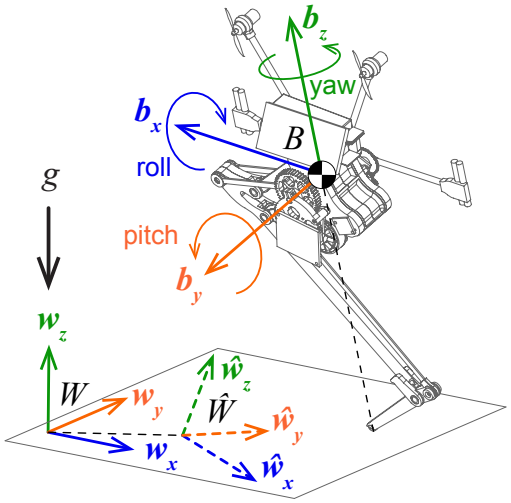

Fig. 2: Robot body, world, and estimated world reference frames.

High-power jumping robots possess neither gravitational references in their free-fall flight nor low stance accelerations in which to correct gyro integration. 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 a solution that estimates and controls robot velocity and angle for a SLIP-like monopedal hopping robot with impulse-like stance phases. This new estimator operates by using 1) stance velocity estimation through an improved dynamic model compared to [18] and subsequently 2) attitude estimate correction from velocity control via touchdown attitude.

II. 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. 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 $\hat{W}$, with a hat.

A. SLIP Hopping Orientation and Velocity Estimator

The following sections detail our SLIP Hopping Orientation and Velocity Estimator (SHOVE). This estimation scheme attempts to correct roll and pitch attitude errors immediately following LO. When the robot detects LO, it makes the following computations:

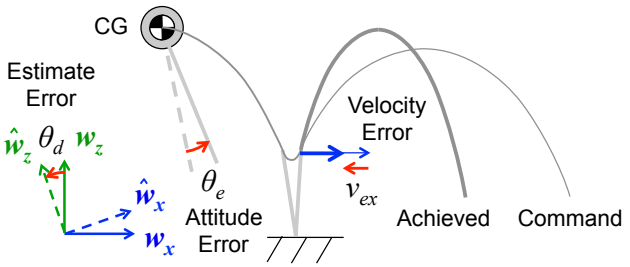

Fig. 3: Attitude errors propagate to liftoff velocity errors so liftoff velocity can be used as an attitude correction signal.

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

B. 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 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 [12] and many proposed subsequently [14] [17] [15] [18].

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 [18] and illustrated in Fig. 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 [18].

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 [18], K_{vx} and K_{vy} are approximately 3 ~~m/s per~~ degree per m/s for Salto-1P [1].

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. [2]

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

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 [2]:

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

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 [18]. 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.

C. 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

[1] K_{vx} & K_{vy} are in deg. per m/s, not m/s per deg.

[2] SHOVE uses estimates for R, $\boldsymbol{v}_{LO}$, v_{ex}, and v_{ey} in $\hat{W}$, not W; hats were omitted at publication. SHOVE implicitly 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. (corrected 4/22/2019 after publication).

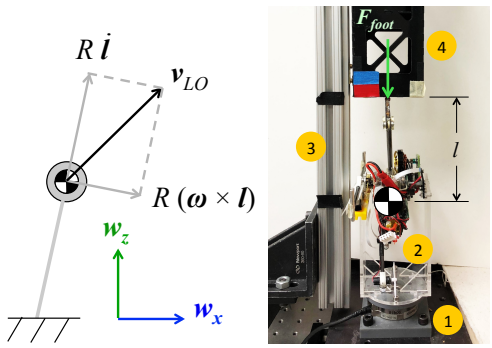

Fig. 4: Left: Calculation of liftoff velocity. Right: Calibration stand: 1) load cell, 2) robot mounted upside-down in load cell carriage, 3) linear slide, 4) cart with red & blue fiducial for tracking.

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:

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

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 $\hat{W}$ [2] 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. 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 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.

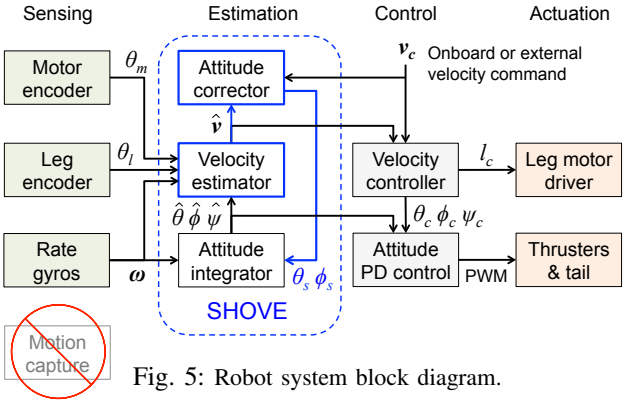

Fig. 5: Robot system block diagram.

D. Velocity Control

Salto-1P controls its hopping height and horizontal velocity using a modified version of the deadbeat foot placement hopping controller developed in [18]. 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 [17], 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.

E. Hardware Platform

We demonstrate SHOVE on the Salto-1P platform developed in [8]. 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 has an MPU 6000 TDK InvenSense 6-axis IMU. Its rate gyroscopes are set to a sensitivity of $\pm$2000 deg/s and its accelerometers are set to a sensitivity of $\pm$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. 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 [6] 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 38 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.

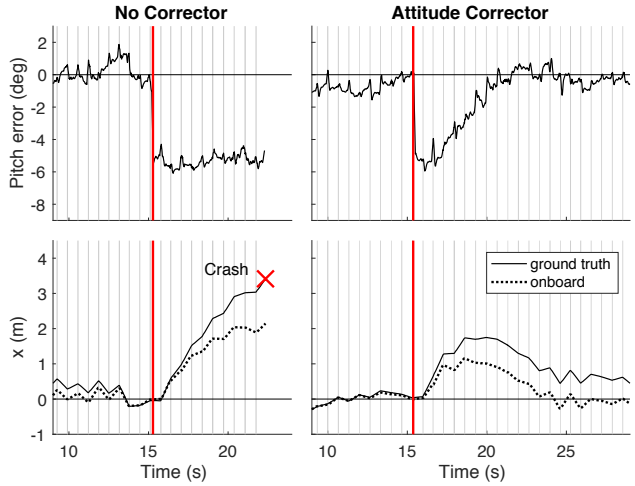

Fig. 6: Disturbance Experiment. At 15.3 seconds (orange line), a software command inserts 5 degrees of pitch error. Without attitude correction, the robot fails to recover and crashes into a wall. With the attitude corrector, the robot's attitude estimate recovers in 10 jumps. Stance phases are shown in grey.

F. 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 intially 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.

III. Results

A. 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. Recorded motion capture data provided a ground truth comparison for the onboard estimates.

At 15.3 seconds, a radio command injected -5 degrees of pitch error into the attitude estimate, shown in Fig. 6. Without

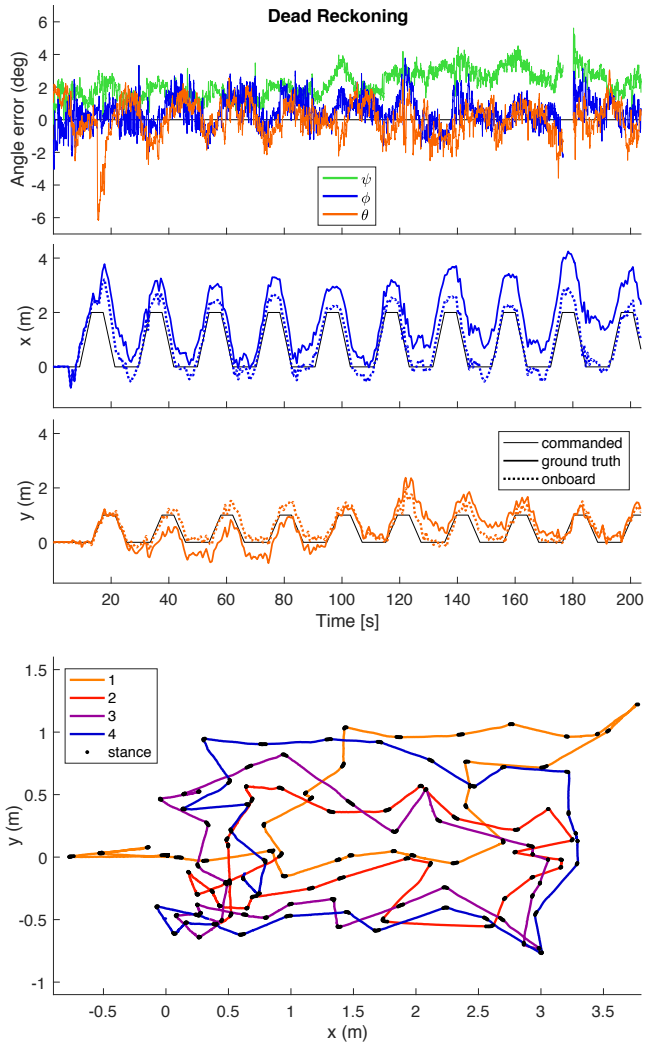

Fig. 7: Dead reckoning experiment hopping along a 2 m by 1 m rectangle. Top: Attitude error. Yaw drifts slightly while roll and pitch remain near 0. The robot exited the motion capture tracking region at 176.7 seconds and re-entered it at 180.1 s. Middle: robot x and y position compared to onboard estimate and command. Bottom: Overlay of first four consecutive rectangles showing position drift.

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.

B. Dead Reckoning

To test the long-term stability of SHOVE attitude estimation, the robot followed a pre-programmed 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. 7. The dead-reckoned onboard position estimate 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.

C. 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. 8.

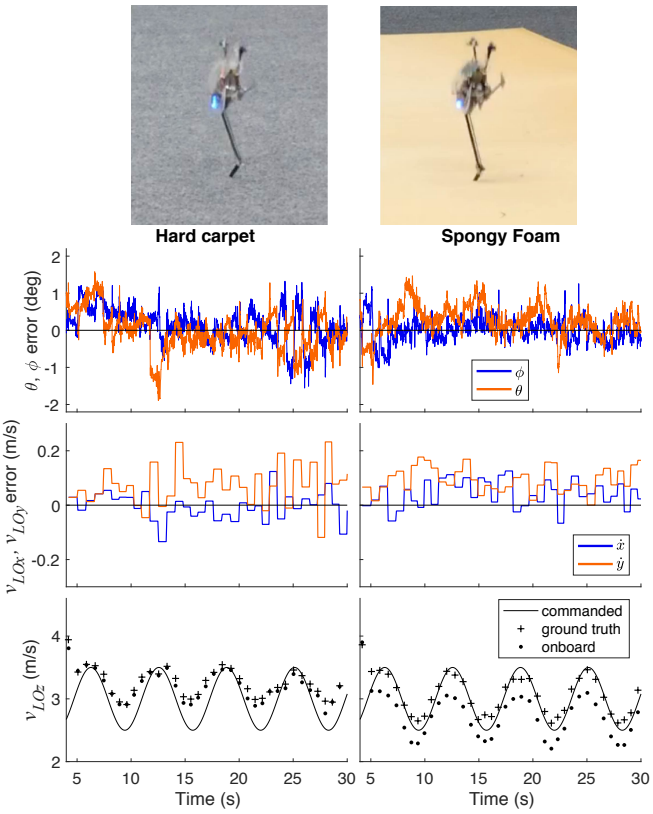

Fig. 8: Experimental comparison on rigid and spongy terrain.

The robot performed well on foam despite a spring constant of 170 N/m for Salto-1P's 5 mm diameter 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 due to the foot's deflection into the foam, the estimator still remained stable and the robot maintained its hopping position on the foam.

D. 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.

Human joystick command allowed the robot to operate in environments 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. 9, the human operator directed the robot to jump in place, and then advance in a straight line towards and onto a step 0.43 m high. The robot was then directed to hop to the right on top of the step, and then towards the left. This run lasted 19 seconds total before human error directed the robot into a bush on the left.

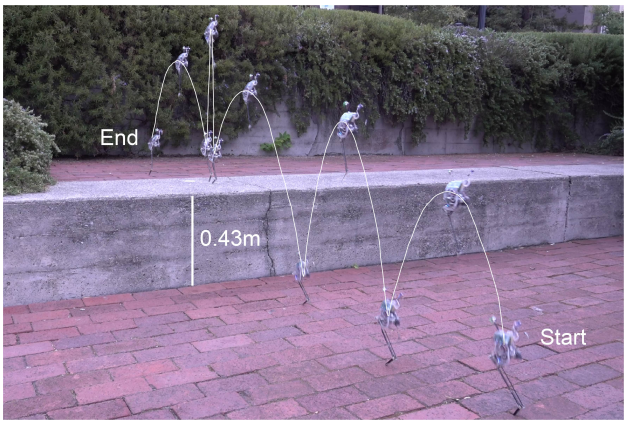

Fig. 9: Fully onboard run under human directed $\boldsymbol{v}_c$.

IV. 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 monopoedal 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.

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. Investgations 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.

Acknowledgement

The authors would like to thank Hersh Sanghvi for assistance in recording outdoor videos and in editing this paper.

REFERENCES

[1] A.-J. Baerveldt and R. Klang, "A Low-cost and Low-weight Attitude Estimation System for an Autonomous Helicopter," Proc. of the IEEE International Conference on Intelligent Engineering Systems, pp. 391–395, 1997.

[2] Z. Batts, J. Kim, and K. Yamane, "Untethered One-Legged Hopping in 3D Using Linear Elastic Actuator in Parallel (LEAP)," in 2016 International Symposium on Experimental Robotics, 2017, pp. 103–112. [Online]. Available: http://link.springer.com/10.1007/978-3-319-50115-4{_}10

[3] R. Blickhan, "the Spring-Mass Model for Running and Hopping," Journal of Biomechanics, vol. 22, no. 1112, pp. 1217–1227, 1989.

[4] H. G. de Marina, F. J. Pereda, J. M. Giron-Sierra, and F. Espinosa, "UAV Attitude Estimation Using Unscented Kalman Filter and TRIAD," IEEE Transactions on Industrial Electronics, vol. 59, no. 11, pp. 4465–4474, nov 2012. [Online]. Available: http://ieeexplore.ieee.org/document/5977026/

[5] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," IEEE Computer Graphics and Applications, vol. 25, no. 6, pp. 38–46, 2005.

[6] R. C. Gupta, "Bhaskara I's Approximation to Sine," Indian Journal of History of Science, vol. 2, no. 2, pp. 121–136, 1967.

[7] D. W. Haldane, M. M. Plecnik, J. K. Yim, and R. S. Fearing, "Robotic vertical jumping agility via series-elastic power modulation," Science Robotics, vol. 2048, no. 1, p. eaag2048, 2016. [Online]. Available: http://robotics.sciencemag.org/content/robotics/1/1/eaag2048.full.pdf

[8] D. W. Haldane, J. K. Yim, and R. S. Fearing, "Repetitive extreme-acceleration (14-g) spatial jumping with Salto-1P," IEEE Int. Conf. Intell. Robots. Syst., pp. 3345–3351, 2017.

[9] K. Jayaram, J. M. Mongeau, A. Mohapatra, P. Birkmeyer, R. S. Fearing, and R. J. Full, "Transition by head-on collision: mechanically mediated manoeuvres in cockroaches and small robots," Journal of the Royal Society Interface, vol. 15, no. 139, 2018.

[10] E. Lefferts, F. Markley, and M. Shuster, "Kalman Filtering for Spacecraft Attitude Estimation," Journal of Guidance, Control, and Dynamics, vol. 5, no. 5, pp. 417–429, 1982. [Online]. Available: http://arc.aiaa.org/doi/10.2514/3.56190

[11] J. G. Nichol, S. P. Singh, K. J. Waldron, L. R. Palmer, and D. E. Orin, "System design of a quadrupedal galloping machine," International Journal of Robotics Research, vol. 23, no. 10-11, pp. 1013–1027, 2004.

[12] M. H. Raibert and J. H. B. Brown, "Experiments in Balance With a 3D One-Legged Hopping Machine," Journal of Dynamic Systems, Measurement, and Control, vol. 106, no. 1, pp. 75–81, 1984. [Online]. Available: http://dx.doi.org/10.1115/1.3149668

[13] H. Rehbinder and X. Hu, "Drift-free attitude estimation for accelerated rigid bodies," Automatica, vol. 40, no. 4, pp. 653–659, 2004.

[14] A. Seyfarth, H. Geyer, and H. Herr, "Swing-leg retraction: a simple control model for stable running," Journal of Experimental Biology, vol. 206, no. 15, pp. 2547–2555, 2003. [Online]. Available: http://jeb.biologists.org/cgi/doi/10.1242/jeb.00463

[15] N. Shemer and A. Degani, "A flight-phase terrain following control strategy for stable and robust hopping of a one-legged robot under large terrain variations," Bioinspiration and Biomimetics, vol. 12, no. 4, p. aa741f, 2017. [Online]. Available: https://doi.org/10.1088/1748-3190/aa741f

[16] J. Van Why, "IMU Integration for ATRIAS," Ph.D. dissertation, Oregon State University, 2016.

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

[18] J. K. Yim and R. S. Fearing, "Precision Jumping Limits from Flight-phase Control in Salto-1P," IEEE Int. Conf. Intell. Robots. Syst., 2018.

[19] G. Zeglin and H. B. Brown, "First Hops of the 3D Bow Leg Hopper," The 5th International Conference on Climbing and Walking Robots, pp. 357–364, 2002.