

Falling with Style: Sticking the Landing by Controlling Spin during Ballistic Flight

Morgan T. Pope¹ and Günter Niemeyer¹

Abstract—As agile robots learn to run and hop, we believe they will soon also perform large jumps and spins that require graceful landing. To land properly, they will need to be aware of their relative body orientation during free fall and make appropriate adjustments.

In this paper, we examine the sensing and actuation necessary for a mechanism to alter its fall under a high rate of spin. We also present a simple new robot, the *Binary Robotic Inertially Controlled brick* (BRICK). This device combines information from an accelerometer, a gyroscope, and a laser range-finder to track its orientation, and adjusts spin mid-flight by rapidly changing its moment of inertia between two states. An error analysis shows mechanical inaccuracies and uncertainty to be the largest source of variation. Nevertheless, our mechanism is able to fall from arbitrary heights, with arbitrary spin, and always land in a desired orientation.

I. INTRODUCTION

As robots are moving beyond the factory and other tightly controlled environments, they have learned to crawl, walk, and even run and occasionally jump. They are becoming more dynamic, not just in their speed of movements but in their ability to handle underactuation, unstable balancing, and intermittent or limited control authority.

Following these trends, this work takes inspiration from exceptional human athletes, such as gymnasts and divers, who are able to stick their landing after multiple fast rotations in the air. They change their body pose at just the right time to alter their spin and bring themselves to the perfect orientation for landing. So we ask: what are the limits of state estimation and control in situations where a robot is thrown through the air with a high rate of spin? How well can we track position and orientation using onboard sensors and what sensors are most suited for the task? What can we do to control the orientation for landing?

To focus our efforts, we study an object which, for now, spins only about the vertical axis, i.e. in the horizontal plane. When dropped from an arbitrary height, at an arbitrary angle, with an arbitrary angular speed, can we always “stick the landing” by matching a desired orientation?

Accomplishing this will, for example, allow a high-aspect-ratio object to drop through a narrow slot or alight on a narrow target platform. If the object is symmetric, then the goal on impact can be achieved by two orientations 180° apart. A schematic of this concept is shown in Fig. 2 and successful implementations of landing on a narrow platform and dropping through a narrow slot are shown in Fig. 1. Success in this task will guide us to more complex

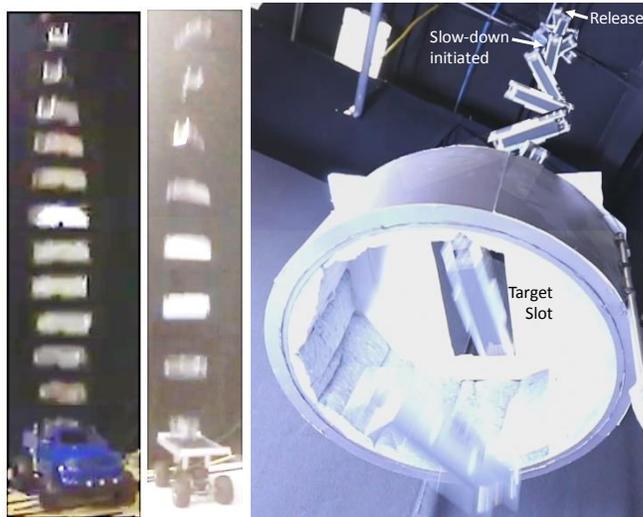


Fig. 1: Left: two image sequences showing the *Bistable Robotic Inertially Controlled brick* (BRICK) spinning through the air and landing in alignment with the beds of two RC trucks. Right: an image sequence taken from below showing the BRICK spiraling down and through a narrow slot, the height and orientation of which were detected using the onboard laser rangefinder in the moments before release.

applications in future work, such as a flipping robot that always lands on its feet. This exploration should also be helpful for fall recovery and for robust terrain navigation. Legged robots should be able to change their moment of inertia mid-flight as we do here, and could use this ability to successfully land after both planned and unplanned leaps.

In pursuit of our initial goal, we look at different ways to change orientation mid-flight in Sec. III and sensing options for tracking position and orientation in Sec. IV. We describe our prototype robot, the *Binary Robotic Inertially Controlled brick* (BRICK, Fig. 3) in Sec. V. The BRICK blends sensing and actuation to get to a goal orientation at the time it hits the ground (Fig. 1). We then analyze the results and measure the sources of error in the robot’s performance.

Selecting this simplified example allows us to explore sensing and actuation strategies in an easy-to-evaluate context, avoiding the complications of full three dimensional jointed locomotion as in [1]. The simplified physics allows us to focus on the initial questions of how to estimate pose and how to effectively modify inertia.

¹Disney Research, Glendale, CA 91201, USA {morgan.pope, gunter.niemeyer}@disneyresearch.edu

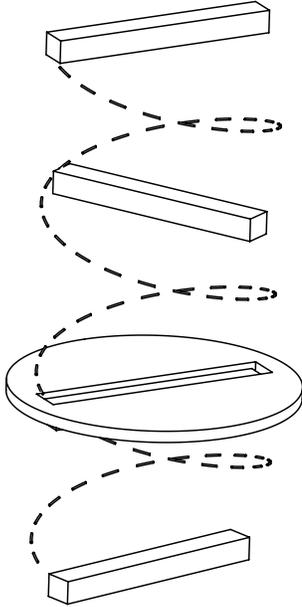


Fig. 2: Our challenge: create an object that spins through the air and achieves a goal orientation at a specific height

II. BACKGROUND

Controlling the orientation of objects is most commonly studied in the aerospace literature, to adjust and regulate spacecraft and satellite attitudes. For actuation, these craft use thrusters [2], flywheels [3], as well as actuated appendages [4].

In robotics, much work has been done on jumping [5] and hopping robots, of which [6] is a foundational example. In general, these robots depend on correct initial jumping conditions to guarantee landing in the proper orientation, although [7] uses a spinning rod to help ensure stability before executing a secondary jump off a vertical surface. The ParkourBot [8] uses gyroscopic stabilization to maintain heading as it performs multiple jumps in a simulated reduced-gravity environment. Other approaches use aerodynamic surfaces to control orientation, as in [9,10]. [11] uses an internal weight to orient correctly while on the ground before using rotors to take off for brief hopping flights.

Changing orientation during free fall has been explored recently by robots which use inertial tails to reorient themselves, a strategy also found in the natural world [12]. By actuating a tail with a relatively high moment of inertia, rapid changes in gross body orientation are possible [13], allowing robots to recover from unfavorable body orientations while falling [14]. Controls for a “cat robot” which used torque between two columnar halves to control final orientation were explored theoretically in [15]. Inertia shaping for a humanoid robot has been explored through in [16]. All of these applications involve relatively low initial spin rates and perform less than a full revolution.

For somersaulting robots, [17] actively shape angular momentum during the ground contact phase and have proposed

a framework for control during a long airborne phase, but have not implemented this strategy because their robot is airborne for such a short period of time. Similarly, [18] characterize angular momentum to inform control while in contact with the ground. Detailed simulations of higher-spin rate humanoid inertia shaping in [19,20] show the potential of this strategy for controlling free-fall landing conditions if it could be realized on a humanoid robot. The most relevant prior work of all is probably [21], in which a somersaulting robot moved its leg position to shape inertia in flight and land at a target angle. The main differences between our approach and that of [21] are that our robot does not change external geometry in effecting its inertia, allowing the effect to be “hidden”, and that we do not rely on an estimation of time in the air based on jumping condition, but instead measure our height at time of release using an onboard sensor. In this paper we explore the sensing and actuation strategies that effectively allow a robot to change its orientation during a ballistic flight with high initial angular velocity and multiple revolutions, without changing external geometry.

III. ACTUATION STRATEGIES

As mentioned above, aerospace applications involve controlling objects in free fall using thrusters, flywheels, and actuated appendages.

For robotics, with faster rotational dynamics and shorter flight times, we focus on fully repeatable mechanical options and do not consider thrusters. Flywheels provide fast and very precise control and are fully reversible. However, they add substantial weight and consume significant power.

Appendages provide an alternative and have been used in jumping robots [12,14]. An inertial tail can create large changes in body orientation in a few tenths of a second [13]. This is useful when it is important for one part of the robot to land in a specific orientation, for instance to present feet towards a surface [14] or to change actuator orientation rapidly. In our example, where we wish to orient the entire body of the robot, the change in gross body configuration is slightly less appealing.

In the following, we take inspiration from human athletes and consider changing moments of inertia. In particular, we shift internal weights perpendicular to the axis of rotation. This can be done by latching weights near the center of mass, then releasing them so they slide out to the ends of the robot, either due to centripetal acceleration or with the assistance of a spring. This requires almost no power, as the only active actuation is releasing the latch. In future work, an actively driven system could be used for the price of higher in-flight power requirements.

Consider the angular momentum, L , which is conserved during ballistic flight:

$$L = I\omega \quad (1)$$

where I is the moment of inertia around the axis of rotation and ω is the angular velocity. If the moment of

inertia changes from I_1 to I_2 , the change in angular velocity is given by:

$$\omega_{new} = \frac{I_1}{I_2} \omega_{old}. \quad (2)$$

Assume that the device is in the air for an amount of time t_{fall} . In order to guarantee an arbitrary landing orientation, the difference in final orientations between never changing inertia and changing inertia immediately must be larger than a full revolution:

$$(\omega_{old} - \omega_{new}) t_{fall} > 2\pi. \quad (3)$$

Or, if the object is symmetric:

$$(\omega_{old} - \omega_{new}) t_{fall} > \pi. \quad (4)$$

This can be rewritten in terms of initial angular velocity, ω , and the different inertias I_1 and I_2 :

$$\omega \left(1 - \frac{I_1}{I_2} \right) t_{fall} > \pi. \quad (5)$$

This shows that there is a minimum speed of spin required to reach an arbitrary goal orientation that depends on both the total time of flight and the size of the inertia change. It is a viable method of controlling final orientation only if that condition is met and if the resulting motion is repeatable and well understood.

This analysis neglects the time needed for actuation and the transition time between different inertias, and so represents the best-case limit. For our device, the BRICK, I_1 is 60% of I_2 , which means that when falling for one second a minimum speed of 2.5π rad/s is required to hit an arbitrary goal orientation.

For a generic jumping robot, time in the air and initial spin rate may be smaller than those used in testing the BRICK; however, it is also probably true that a jumping robot is designed to land in approximately the correct orientation, and so the necessary adjustments are much smaller in magnitude.

IV. SENSING STRATEGIES

We sort our sensing approaches into internal sensors (accelerometers, gyroscopes, and magnetometers) and external sensors (barometers, ultrasonic, and laser range finders). Limitations on internal sensors can be overcome by adding external sensors.

A. Internal Sensors (IMU)

We use the ST LSM6DS33 combined 3-axis accelerometer and 3-axis gyroscope and the ST LIS3MDL 3-axis magnetometer to acquire information about pose. Each sensor has its strengths and weaknesses in light of the current challenge.

The gyroscope on the ST LSM6DS33 has a measured noise floor of $\pm 0.15^\circ/s$ when sampled at approximately 800 Hz. When at rest, the heading derived from the gyroscope readings drifts noticeably after a few tens of seconds

of integration. Also, when spinning it becomes crucial to precisely calibrate the scale of the gyroscope in each axis, as even very small errors in angular velocity rapidly accumulate into large angular position errors. Our current calibration allows us to track orientation at several Hz with a little more than 1° of error accumulated per revolution.

When at rest, the accelerometer can be used to resist drift in the estimated orientation. The accelerometer precisely identifies the vertical axis, and can reliably detect the onset of free fall and the impact with the ground. If the accelerometer is not located at the center of mass of the object, care must be taken to cancel out centripetal accelerations when spinning.

Once in free fall, the accelerometer provides very little usable information, as drag forces on the body tend to be negligible. We instead depend solely on the gyroscope. Since the time of fall is on the order of a second, significant drift does not accumulate in the gyroscope heading during flight.

We note that it should be possible to integrate the accelerometer reading to provide an estimate of motion, but integration errors accumulate after a few seconds (and centripetal accelerations must be very carefully canceled). Thus, in future work we hope to use the accelerometer to measure acceleration during a launch from ground level and predict flight time; for now we must rely on external sensors to measure vertical position.

A magnetometer should, in theory, also be used to resist drift both at rest and in motion. In an environment where the earth's magnetic field is consistent and larger than other sources of electromagnetic noise, the magnetometer can provide heading information in the horizontal plane. The passive actuation method chosen means that local electromagnetic noise is relatively small. However, in a building full of large ferrous objects and powered machinery, we found the global magnetic signal to be so inconsistent both spatially and temporally as to be unusable.

We blend these sensors by using the accelerometer to correct for drift in the pose estimate while not in free fall. We use a .15 Hz low-pass filter on the accelerometer readings in performing this correction. This allows us to have reasonable confidence in our estimated orientation in the moments before beginning the experiment.

B. External Sensors

Determining the time of flight requires a knowledge of vertical height and initial velocity, both of which are observable by external sensing. The following experiments all assume zero initial velocity, so we focus on determining vertical height.

In attempting to ascertain height, we explored using the ST LPS25H barometer and the HC-SR04 ultrasonic sensor before settling on the Lidar-LITE v.1 laser rangefinder. Using a barometer proved problematic because interpreting measurements required knowledge of ambient pressure that changes with weather conditions. Additionally, the data from the ST LPS25H was both too noisy and too coarse for our purposes.

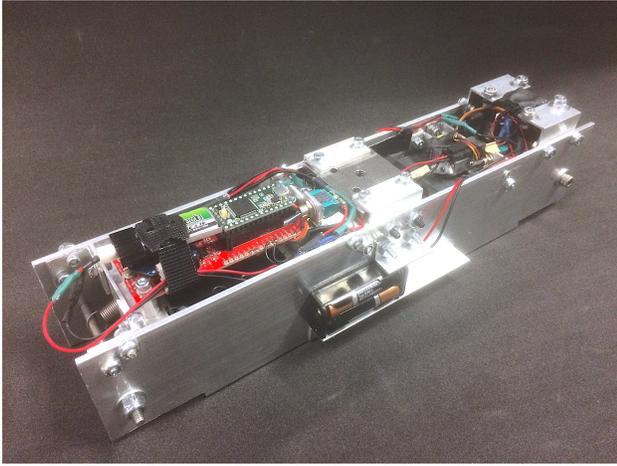


Fig. 3: The *Binary Robotic Inertially Controlled brick* (BRICK)

Both ultrasonic and laser rangefinders were deemed appropriate for measuring height over several meters at tens of Hz. However, the tight beam of a laser allows for a secondary sensing goal to be achieved: determining the orientation of a target landing zone. The beam from the Lidar-Lite v.1 is a cone with an angle of 3° , allowing it to see through small gaps at a distance of several meters. If the range finder is placed on the extreme end of the robot, then when the robot is rotated over a flat horizontal surface with a narrow slot in it, the range finder will sometimes be directly over the slot and sometimes directly over the surface. Thus the robot can both detect the height of the target surface and the angle of the slot in the robot's reference frame. This allows us to adapt to the environment by detecting the orientation of a slot or platform.

V. PROTOTYPE

We built the BRICK (Fig. 3) to test specific sensing and actuation choices. In this section we will describe the mechanical design of the robot and then discuss the choices made for sensing and actuation.

The robot is built around two 650 g rectangular steel weights that move along a linear slide. A cable system forces them to mirror each other's movements, and steel springs pull them outwards. The weights and the slide are protected by an aluminum shell. Also supported by the shell are the sensing, computation, and communication pieces of the robot. The robot uses the ST LSM6DS33 and the LIDAR-Lite laser rangefinder to detect its position and orientation, a Teensy 3.1 microprocessor for logic and control, and an Xbee radio for reporting state variables back to a computer for logging. A Hitec HS-45B servo is mounted so that the plane of rotation of its servo arm is parallel to the outside edge of one of the steel weights. When the arm is rotated down, the outside edge of the steel weight rests against it, preventing it from flying out to the end of the robot. The cable system guarantees that if one steel weight is stationary, the other cannot move either. At the appropriate time, the arm rotates up so that it clears the top of the steel weight, allowing both weights to

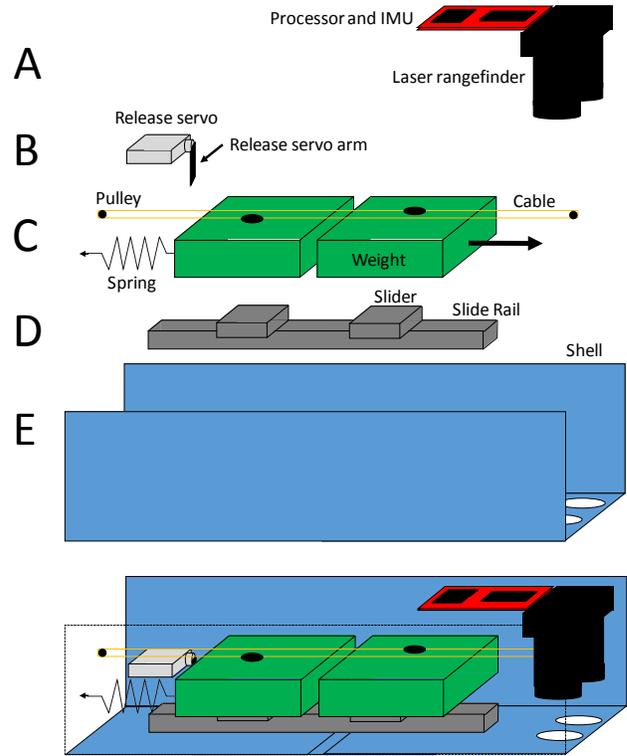


Fig. 4: Top: exploded schematic of prototype components: A) sensing (imu and laser rangefinder), computation (microprocessor), and communication (radio); B) release servo; C) weights, cable, and spring; D) linear slide; E) protective shell. Bottom: cutaway view of assembled components.

spring out to the distal ends of the robot. These components are illustrated in Fig. 4. The full assembly weighs 2.6 kg.

A piece of steel with a threaded hole is attached to the top of the robot and lined up vertically with the center of mass. This lets us screw the robot onto a 1/4-20 bolt affixed to the ceiling of the laboratory. When the robot is spun clockwise, it releases from the bolt and falls to the ground. To prevent damage to the robot during experimental trials, a large foam mat was placed below to reduce impact forces. Velcro was attached to both the bottom of the robot and the mat to minimize rebound.

To drop the BRICK, the robot is first screwed onto the ceiling-mounted bolt, then spun by hand at a rate between 1.5 and 4 Hz. Since the screw is fixed, the robot leaves the screw in roughly the same orientation every time. To make sure this was not biasing the results, the screw was rotated ninety degrees halfway through testing, with no noticeable difference in results. To guarantee that the robot falls straight down, it is important to exactly line up the threaded hole with the robot's center of mass.

The prototype is shown in operation in the multimedia attachment accompanying this paper.

VI. FRAMEWORK FOR SENSING AND CONTROL

We combine data from the accelerometer, the gyroscope and the laser range finder to inform when to actuate the

release servo.

On startup, the robot assumes that it is at rest and uses data from its accelerometer to determine which way is up. The robot's orientation on power-up is assumed to be the zero position in the horizontal plane.

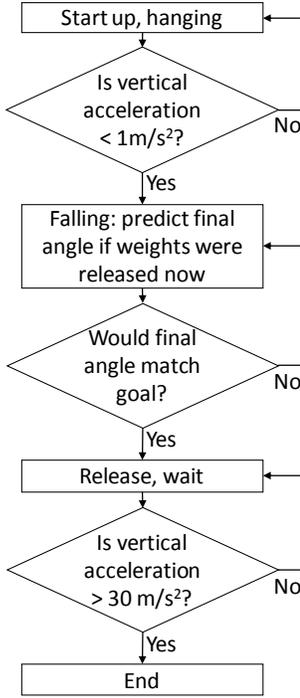


Fig. 5: Flowchart showing the logical framework for the BRICK's state machine.

We track orientation as a quaternion, q , which is updated by the gyroscope reading, ω , at every time step:

$$\frac{d}{dt}q = \frac{1}{2} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \\ 0 \end{bmatrix} \otimes q \quad (6)$$

When the BRICK is at rest, we use the data from the accelerometer to compensate for drift in our estimated orientation. This is done by taking the cross product of the internal estimate of vertical, z_i , with the accelerometer heading, z_w , to create a correction term, ω_a , that is equal to the error caused by the drift:

$$\omega_a = z_i \times z_w. \quad (7)$$

This ω_a is then scaled by a constant, λ , to filter out high-frequency noise. The scaled result is subtracted from the gyroscope's measured ω to make the adjusted value, $\omega_{adjusted}$:

$$\omega_{adjusted} = \omega - \lambda\omega_a. \quad (8)$$

This is then used to update the q as in Eq. 6.

The state machine that controls the logic of the BRICK is illustrated in Fig. 5. When the robot sees the acceleration in the vertical axis (smoothed using a 20 Hz averaging filter) drop below a threshold, the algorithm reviews logged vertical acceleration data to find out when the raw accelerometer reading dropped below the threshold for free fall. It then uses the height measurement from the laser rangefinder from that time step to estimate the total falling time, T_f :

$$T_f = \sqrt{\frac{2h}{g}} \quad (9)$$

where h is the measured height and g is acceleration due to gravity.

The state machine then uses the following formula to estimate the final orientation, θ_{final} , if the release were to be commanded immediately:

$$\theta_{final} = \theta + \omega T_r + \omega R_t T_w + \omega R_f (T_f - T_r - T_w - t) \quad (10)$$

where θ is the current estimated angle in the horizontal plane, ω is the current angular velocity about the vertical axis, T_r is the time required for the servo to release the weights, T_w is the time required for the weights to slide to the end of their travel, R_t is the ratio between the current angular velocity and the average angular velocity while the weights are sliding, R_f is the ratio between the current angular velocity and the final angular velocity after the weights have reached the end of their travel, T_f is the total fall time calculated from the initial height measurement, and t is the time elapsed since entering free fall.

When θ_{final} goes from just before the target position to just after it, the command is sent to raise the servo arm and release the weights. After the servo arm is set to the correct position during startup, the servo does not receive any signals from the processor until this command is sent. This is to make sure that the servo responds right away instead of waiting for its own 50Hz update loop to complete.

VII. RESULTS AND ANALYSIS

Fig. 6 shows the passive behavior of the device when dropped at arbitrary angular velocities from a fixed height. Fig. 7 shows the effect of the BRICK's control and actuation strategy. The passive drops result in random final orientations, while the addition of control produces consistent headings. A typical set of acceleration and angular velocity measurements during an experiment is shown in Fig. 8.

The final prototype missed the goal orientation by an average of 1° with a standard deviation of $\pm 9^\circ$ over ten trials. This error is assumed to be caused by drift and noise in the sensors and by variability in the physical system. Four sources produced most of the variability in the system, two mechanical and two caused by sensor error. In Fig. 11 we quantify the effect of each of these error sources: the time needed to release the weights, the motion of the weights, the height measurement and the accumulated inaccuracies in

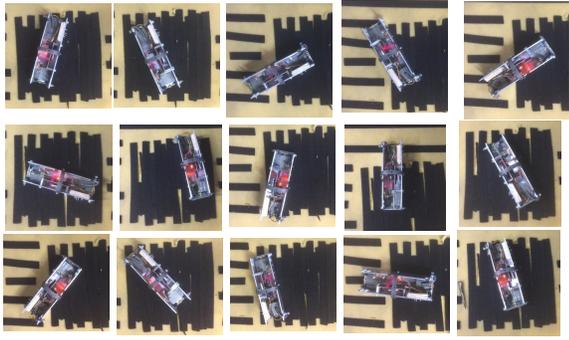


Fig. 6: Final orientations without predictive landing control.

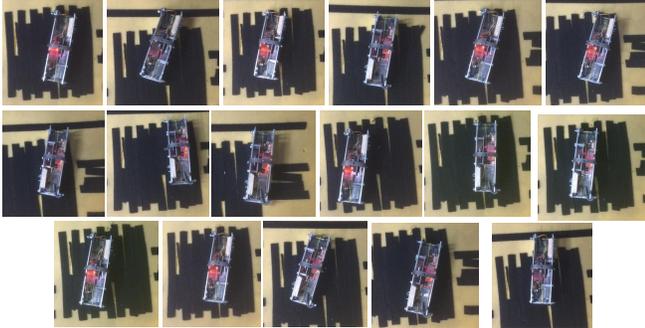


Fig. 7: Final orientations with predictive landing control.

heading sensing and estimation. For each source, we measure the individual variations and convert this into angular displacements. We then plot the average bias resulting from these factors, with bars that show the standard deviation, to give us a sense of the relative variability caused by each error source. Each error source is discussed in more detail in the subsections below.

A. Mechanical Error Source: Release Actuation

The weights are held in place by a small servo arm. When release is commanded, the time at which the weights are actually free to move varies due to factors like friction, servo battery level, servo firmware, and angular velocity. The exact time that the weights begin to move is difficult to estimate, but is usually accompanied by a small spike in the acceleration data. This time is then converted to an estimated angular error by multiplying the difference between the actual and the expected time of actuation by the average angular velocity and plotted in Fig. 11. To reduce the error associated with this factor, the authors halt code execution at release time to send the appropriate pulse instead of using a background library function to drive the servos. Despite this and the relatively low-friction plastic surface of the servo arm, release actuation is one of the largest sources of variability.

B. Mechanical Error Source: Motion of Weights

If the weights are not driven by a spring, the forces exerted on them will be proportional to angular velocity. The springs used in our prototype dominate but do not eliminate this effect. Friction along the slider is another source of variability in this case, as is stretch in the cable

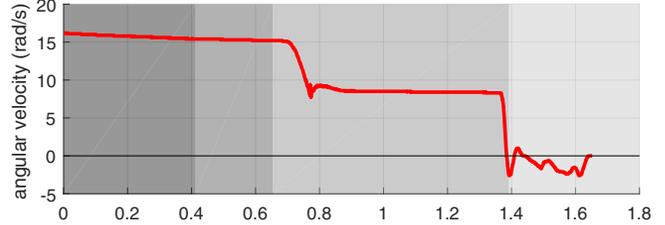
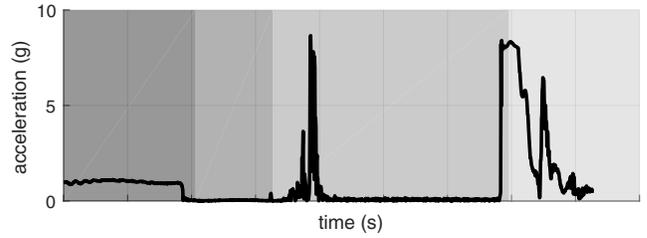


Fig. 8: Typical vertical acceleration and angular speed during a controlled fall. Shading indicates machine state, from darkest to lightest: Start-up, Falling, Release, End; see Fig. 5.

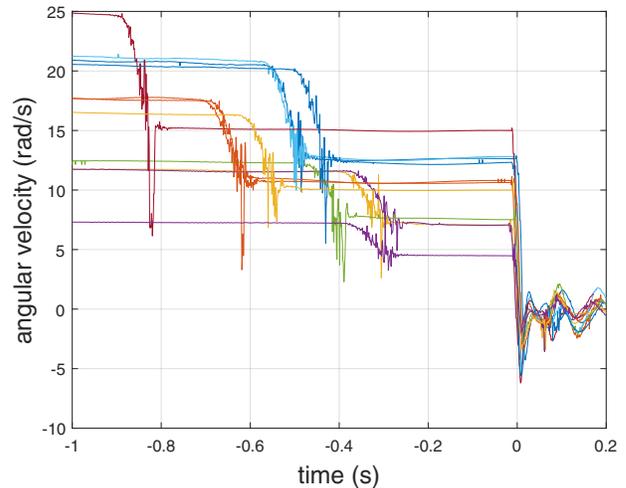


Fig. 9: Measured angular velocity for ten successive experiments. Initial angular velocity ranges from 7.5 to 25 rad/s. The drop in angular velocity occurs as the weights are released and the inertia of the system increases. On impact, angular velocity drops to zero with some ringing.

connecting the two weights. If the variation was primarily caused by changing angular velocity, it could be compensated for in software. However, the above difficulties in finding out exactly when the weights begin to move make it difficult to parse how much of the variation is caused by differences in angular velocity and how much of it is caused by friction. Fig. 12 shows the consistency of actuation achieved so far by overlaying ten series of angular velocity measurements in the quarter second after actuation is commanded. Variability in the motion of the weights is the single largest source of variation (Fig. 11).

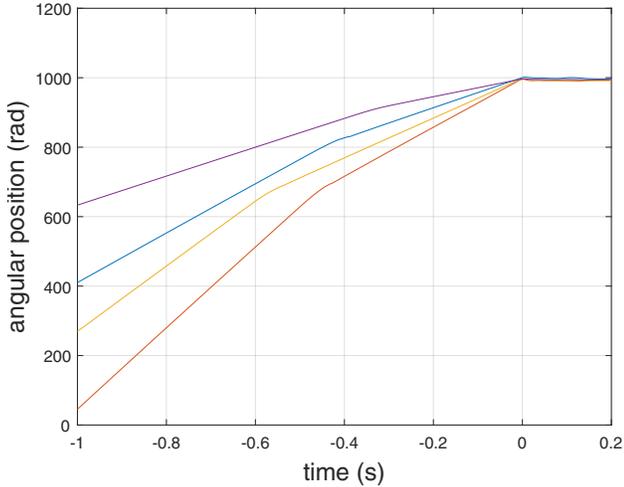


Fig. 10: Measured angle for four select experiments, referenced from the target landing orientation, showing variety of paths to convergence. Each experiment starts with a different offset angle, but converges to the target landing orientation. The knee in each graph indicates when the angular velocity decreases because of the release of the weights.

C. Sensor Error Source: Height Measurement

The height sensor provides an estimation of starting height once free fall is detected, which can then be used to estimate the time spent in the air. This can then be compared to the actual time of flight as measured by observing the accelerometer spike caused by impact. The difference in time multiplied by the average velocity gives us an estimated error from this measurement as shown in Fig. 11.

We use the initial measurement and predict height from there based on simple dynamics, neglecting drag. We could update our estimate during flight by comparing our model to new measurements, but in practice we do not see a strong need to do this since the model predicts the data closely. Two typical sets of height measurements are plotted in Fig. 13, along with the values predicted by the model.

D. Sensor Error Source: Pose Estimation

When the robot impacts the ground, its actual pose differs from its internal calculated orientation. The difference between actual and internal pose is a measurement of sensor bias and is plotted in Fig. 11. This bias can result from scaling issues and from at-rest offset. We are fairly confident that the offset is small because the robot is left at rest for a second on startup to calibrate the offset and will not operate if it detects too much variability in the gyro signal (indicating motion). The source of this bias is probably small errors in scale calibration. The variability of this bias comes from drift in the gyro and in the integration of the measurements over time.

E. Design lessons

The preceding sections show that error comes from both mechanical design and sensing inaccuracies. The largest

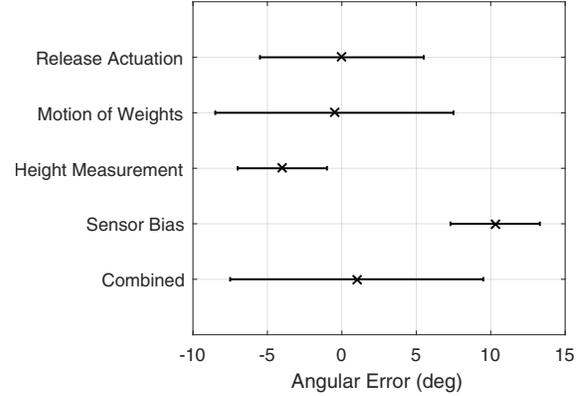


Fig. 11: Estimated variability caused by four principal error sources, with final error in target pose at far right. Error bars represent standard deviation from the mean. The magnitude of each error should be correctable in software by updating constants. Reducing the variability in each error source requires some combination of better mechanical design, more complicated modeling, or better sensors.

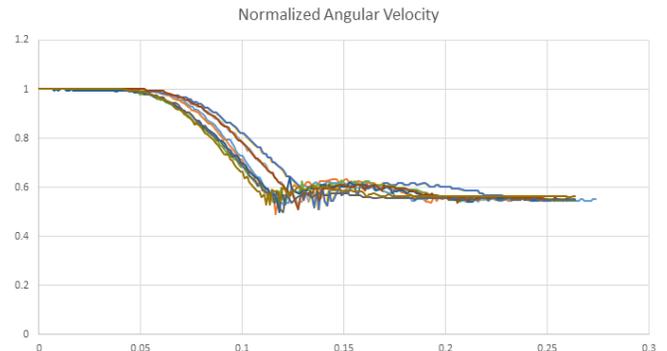


Fig. 12: Overlay of normalized angular velocity versus time for ten consecutive trials showing the repeatability of the spring-driven transition from one configuration to another.

variability comes in fact from mechanical issues: the release of the servos and the motion of the weights. In order to achieve precise angular results when falling at high spin rate, it is crucial that the mechanical system be made as repeatable as possible. In this prototype we have done our best to minimize unpredictable factors, especially friction along the linear slide and between the servo and the weights, but better performance requires further improvements.

The current control scheme is unforgiving to even the slightest variation in physical response. If the weights were released continuously by a motor or variable brake instead of all at once by a single binary input, there would be more opportunities to compensate for mechanical variability, and sensing errors might become the larger source of variation.

VIII. CONCLUSION AND FUTURE WORK

As robots move in more dramatic and dynamic ways, the need for tracking and modifying in-flight orientation will become more common. We have presented a simple

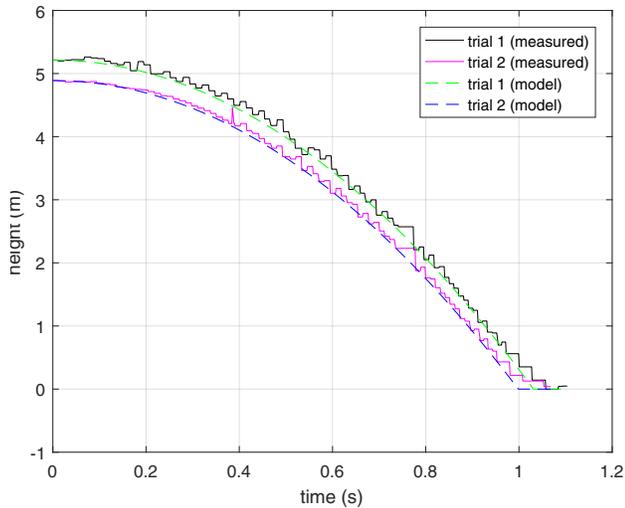


Fig. 13: Modeled versus measured data for a typical fall. The model matches the measurements closely but with less noise.

mechanical system that is capable of changing inertia with the high level of repeatability required for reaching a target orientation. We combine information from accelerometers, gyroscopes, and a laser range finder to estimate pose and note the noise introduced by each.

In future work, we will explore how best to balance complexity in achieving our desired goals. The actuation strategy chosen for the BRICK requires very little power to operate, but places tight constraints on repeatability. A more active system with higher power draw, for example continually adjusting the moment of inertia, might relax these constraints while shifting complexity to the controller.

Future goals also include operating with a horizontal spin axis. This presents challenges in separating inertial accelerations from gravity and in detecting height while rotating towards and away from the ground plane. Such flipping motions, however, could lead to dramatic robotic capabilities. Ultimately we hope to understand and control orientation as the robot freely tumbles and falls.

It should be noted that in this work we are attempting to change the landing angle by as much as 180° , and hence the required spin rate and fall time are substantial. A jumping robot might have less time and spin rate to work with during a typical jump, but we also expect that such a robot would be trying to correct errors in orientation that are much smaller than 180° . It is partly because of this consideration that we believe our method can be useful for applications beyond the simple demonstration shown in this paper.

The high performance and ever decreasing cost of MEMS and lidar sensors continues to enable new applications. This paper explores one exciting path and, we hope, will help push robots to achieve new and amazing capabilities.

REFERENCES

- [1] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, no. 2-3, pp. 161–176, 2013.
- [2] G. C. Walsh, R. Montgomery, and S. S. Sastry, "Orientation control of the dynamic satellite," in *American Control Conference, 1994*, vol. 1. IEEE, 1994, pp. 138–142.
- [3] P. Tsiotras, H. Shen, and C. Hall, "Satellite attitude control and power tracking with energy/momentum wheels," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 1, pp. 23–34, 2001.
- [4] L. J. Kamm, "'vertistat': An improved satellite orientation device," *Ars Journal*, vol. 32, no. 6, pp. 911–913, 1962.
- [5] R. Armour, K. Paskins, A. Bowyer, J. Vincent, and W. Megill, "Jumping robots: a biomimetic solution to locomotion across rough terrain," *Bioinspiration & biomimetics*, vol. 2, no. 3, p. S65, 2007.
- [6] M. H. Raibert, *Legged robots that balance*. MIT press, 1986.
- [7] D. W. Haldane, M. Plecnik, J. Yim, and R. Fearing, "Robotic vertical jumping agility via series-elastic power modulation," *Science Robotics*, vol. 1, no. 1, p. eaag2048, 2016.
- [8] A. Degani, S. Feng, H. B. Brown, K. M. Lynch, H. Choset, and M. T. Mason, "The parkourbot—a dynamic bowleg climbing robot," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 795–801.
- [9] M. Kovač, O. Fauria, J.-C. Zufferey, D. Floreano, *et al.*, "The epfl jumpglider: A hybrid jumping and gliding robot with rigid or folding wings," in *Robotics and Biomimetics (ROBIO), 2011 IEEE international conference on*. IEEE, 2011, pp. 1503–1508.
- [10] A. L. Desbiens, M. T. Pope, D. L. Christensen, E. W. Hawkes, and M. R. Cutkosky, "Design principles for efficient, repeated jumpgliding," *Bioinspiration & biomimetics*, vol. 9, no. 2, p. 025009, 2014.
- [11] E. Beyer and M. Costello, "Measured and simulated motion of a hopping rotochute," *Journal of guidance, control, and dynamics*, vol. 32, no. 5, pp. 1560–1569, 2009.
- [12] T. Libby, T. Y. Moore, E. Chang-Siu, D. Li, D. J. Cohen, A. Jusufi, and R. J. Full, "Tail-assisted pitch control in lizards, robots and dinosaurs," *Nature*, vol. 481, no. 7380, pp. 181–184, 2012.
- [13] C. Casarez, I. Penskiy, and S. Bergbreiter, "Using an inertial tail for rapid turns on a miniature legged robot," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5469–5474.
- [14] A. M. Johnson, T. Libby, E. Chang-Siu, M. Tomizuka, R. J. Full, and D. E. Koditschek, "Tail assisted dynamic self righting," in *Proceedings of the Fifteenth International Conference on Climbing and Walking Robots*, July 2012, pp. 611–620.
- [15] Z. Weng and H. Nishimura, "Final-state control of a two-link cat robot by feedforward torque inputs," in *Advanced Motion Control, 2000. Proceedings. 6th International Workshop on*. IEEE, 2000, pp. 264–269.
- [16] S.-k. Yun, A. Goswami, and Y. Sakagami, "Safe fall: Humanoid robot fall direction change through intelligent stepping and inertia shaping," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 781–787.
- [17] S.-H. Hyon, N. Yokoyama, and T. Emura, "Back handspring of a multi-link gymnastic robot—reference model approach," *Advanced Robotics*, vol. 20, no. 1, pp. 93–113, 2006.
- [18] B. Ugurlu and A. Kawamura, "On the backward hopping problem of legged robots," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 3, pp. 1632–1634, 2014.
- [19] S. Ha, Y. Ye, and C. K. Liu, "Falling and landing motion control for character animation," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, p. 155, 2012.
- [20] J. T. Bingham, J. Lee, R. N. Haksar, J. Ueda, and C. K. Liu, "Orienting in mid-air through configuration changes to achieve a rolling landing for reducing impact after a fall," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 3610–3617.
- [21] R. R. Playter and M. H. Raibert, "Control of a biped somersault in 3d," in *Intelligent Robots and Systems, 1992., Proceedings of the 1992 IEEE/RSJ International Conference on*, vol. 1. IEEE, 1992, pp. 582–589.