Toward Robust Stair Climbing of the Quadruped using Proprioceptive Sensing

Zhiyi Ren[1] and Aaron Johnson[1]

***Abstract*—Uneven stairs and ground obstacles often cause legged robots to trip and even fall over. Autonomous traversal requires robots to detect leg disturbances accurately and react in time. We present an approach that combines a leg observer for contact detection with reactive behavior for recovery. Exploiting the mechanical transparency of the robot design, the observer achieves leg proprioception and contact identification with minimal delay. Reactive algorithms then trigger the robot to switch between walking and stair-climbing, or to adjust swing leg trajectories to step over obstacles. We demonstrate the reliability and potential of the contact observer on robust stair climbing. We envision that future work will establish an autonomy framework for legged robots to traverse through multi-component natural terrains using similar proprioceptive sensing strategies and reactive behaviors.**

I. Introduction

Legged robot can traverse through various difficult terrains that obstruct traditional wheeled or tracked robots, benefiting from large leg workspace and contact geometry reconfiguration [1]. However, as one of the most common obstacles in both indoor and outdoor environments, stairs remain challenging for legged robots to climb over. Due to the inherent geometry of normal stairs, small-sized legged robots often need to use more dynamic gaits such as jumping and bounding [2], and compositions of these dynamic behaviors often risk the stability of the robots. Timely transitions between behaviors require precise detection of the stair tread. In addition, any unexpected disturbances on the stairs may cause the robots to lose balance and even fall over, demanding quick detection and recovery.

For robust autonomous stair climbing, we believe that exteroception with camera and other sensors is essential to the overall system in the future. But robots cannot entirely rely on the the visual approach: small bumps on the stairs may be difficult for the camera to notice, and the camera view is often occluded. Computer vision algorithms are yet to be developed to accommodate the highly dynamic motion of legged robots, with the state of the art likely to generate insufficiently precise stair location data. Meanwhile, proprioceptive sensing provides a more intuitive and reliable solution to contact and disturbance detection. Using motor encoders and IMU built-in on the robot, it can "feel" the object and perceive its orientation as it walks through terrains. It is usually computationally inexpensive, thus reducing the detection latency and allowing for in-time recovery.

Our vision is to let the robot traverse through any type of stairs in any environment. Most of the existing stairs are designed specifically for human walking, and the relative size of the stair may pose a problem for the robot. Our earlier attempts showed that the high slope of some stairs consistently led to loss of stability during behavior transitions. Furthermore, outdoor stairs are often rugged and uneven, even sometimes tripping humans. To simulate a similar setting in an indoor, controlled environment, we chose a set of stairs with a smaller slope (about 30 degrees), and place obstacles on the stairs. The robot is expected to trot towards the stair, detect and ascend, and recover from disturbances.

Fig. 1. Minitaur robot on the stairs used for testing.

Here we present the beginning of a solution to robust stair climbing of the robot. The gaits used include trotting on the ground for approaching the stairs, trotting on the stairs for aligning with the next one, and bounding of the front and rear legs alternatively for ascending the stairs. Leg state estimation is achieved with both offline simulation and online observers, which only require motor shaft encoders as the onboard sensor. As the robot walks towards the stair with a pre-defined leg trajectory, the actual leg positions from the encoders are compared to the outputs of the offline simulation, forming the residual values. A significant residual value during the expected air phase indicates the leg contact with the first stair, and signals the robot to start bounding up. Meanwhile, upward bounding of the legs does not follow a specific leg trajectory, and a online observer is necessary to generate expected leg states. High residual values indicate either the leg touchdown or the presence of obstacles at the legs, which quickly adapts to a certain trajectory for recovery.

The paper is organized as follow: the following subsection

[1]Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213.

presents major related works in robot stair climbing, and some background and justifications of the robot used in our studies. Section II introduces the proprioceptive sensing in more details, explaining the offline simulation and online observer in the use of disturbance detection. Section III demonstrates the gait compositions for stair climbing, and the reactive behaviors for disturbance recovery. Section IV documents the experimental results of the detection algorithm and reactive behavior. The paper finishes with a conclusion and multiple future directions.

A. Related work

Stair climbing has been researched with different types of robots and strategies. Tracked robots enjoy enhanced stability over traditional wheel robots thanks to greater ground contact area and lower center of gravity. Xiong and Matthies [3] demonstrated using computer vision algorithms to locate the stair edges and determine the relative orientation of the stair and robot for guidance. Steplight et al. [4] developed a hierarchical stair-climbing model using sensor fusion for a tracked robot, and work in [5] improved the robustness of the sensor fusion approach by introducing extended Kalman filtering for state estimation. Meanwhile, few legged robots have been used in stair climbing. One of the earliest versions of the RHex hexapod robot exploits a special curved leg design [6] for stair clmbing. The RHex robot further incorporated stair sensing and sequential compositions of controllers to achieve autonomy [7], [8].

However, almost all of these works demonstrated the climbing strategies on regular indoor stairs with no obstacles. Some of the works [5] dealt with collisions with stairs but not any unexpected irregularities of the stair shape. It is certain that most of the algorithms would fail in an outdoor rugged environment, where the stairs may be littered with small obstacles.

Fig. 2. Outdoor stairs at Schenley Park, Pittsburgh, PA.

We addressed the problem with a quadrupedal platform, the Minitaur from Ghost Robotics (Fig. 1). Each leg is lightweight (Table 1) and driven by two DC brushless motor with no gearbox. This direct-drive design results in high mechanical transparency [9], which enables high leg acceleration and ground impulse detection in minimal delay. It has been demonstrated that the Minitaur is capable of a series of stable, dynamic maneuvers including bounding and trotting [10]. We believe that the high maneuverability and high mechanical transparency render the robot a competent choice for stair climbing. It is also equipped with motor encoders and an IMU, and the STM32 ARM microcontroller onboard updates motor commands at a maximum of 1 KHz subject to the amount of computations.

One issue that we encountered was that the leg length of Minitaur is almost the same as the height of normal-sized stairs, which causes difficulty in walking up the stairs. Topping et al. [2] documented the quasi-static mismatch between Minitaur and normal-sized stairs due to the leg size and torque limit, introducing a dynamic, pronking-like gait for stair ascent. However, we believe that this gait is vulnerable to any disturbance on the stairs. With all four legs in the air and hence no ground contact when any collision occurs, the robot is likely to lose stability and fall over on the stairs. Instead we adopt a less dynamic, bounding-like gait, with front and rear pairs of legs ascending alternatively. While the other pair of legs is anchored to the floor, the legs in the air are able to quickly adjust the trajectory when contacting obstacles, maintaining whole-body stability.

While we intend to develop a complete stair climbing behavior for the Minitaur, our focus in this paper, as the first step towards robust traversal over uneven stairs, is to introduce proprioceptive sensing for obstacle detection. Disturbance observer has been applied on multiple legged robot platforms, including the RHex robot [11] and some humanoids [12]. While many other novel approaches, such as using particle filtering [13] or probabilistic contact fusion [14] have emerged, we believe that a simple, deterministic observer model is sufficient to spot the changes in leg dynamics caused by the obstacles. Also, the simple model is less demanding in computations for the sole onboard microcontroller. As the target terrain becomes more complicated, we would improve upon the simple model in the future.

II. Proprioceptive Sensing

The general approach is to simulate the 2D leg motion in finite time steps. We ignore the contact force and focus on the period when the legs are in the air. Significant deviations from the expected states (high residuals) indicate that the leg has "felt" a stair or obstacle.

A. Leg Dynamics

The Minitaur robot has a symmetric five-bar mechanism for each of the four legs. Each leg is driven by two brushless DC motors to move in the sagittal plane. The kinematics of the mechanism is detailed in [9]. We assume that the center of mass of each link is located at the geometric center.

We use Euler-Lagrange equations to solve the leg dynamics. The generalized coordinates are $\theta \in \mathbb{R}^{2\times 1}$, the two motor angles. Consider the following equation of motion.

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) = \tau \tag{1}$$

where $M \in \mathbb{R}^{2\times2}$ is the mass matrix obtained by combining the mass each of the four link in a leg, $C \in \mathbb{R}^{2\times2}$ is the Coriolis matrix, $N \in \mathbb{R}^{2\times1}$ is a vector that contains gravity and joint friction terms, and $\tau \in \mathbb{R}^{2\times1}$ is the torque vector from the ideal motor model based on the applied PWM,

$$\tau = K_t I \tag{2}$$

$$I = \frac{0.95 \cdot V \cdot PWM - V_{emf}}{R} \tag{3}$$

$$V_{emf} = K_t \dot{\theta} \tag{4}$$

where K_t is the torque constant (and the back electromotive force (emf) constant), V is the supplied voltage measured by onboard sensor, R is the armature resistance. The 5% voltage deduction is built-in and confirmed with Ghost Robotics. A standard proportional-derivative (PD) controller determines the motor PWM value as follows,

$$PWM = K_p(\theta_d - \theta) + K_d(0 - \dot{\theta}) \tag{5}$$

where K_p and K_d are the proportional and derivative gains, and θ_d is the target position. Note that we do not set a reference velocity here. The controller gains are tuned for stable behavior.

B. Model Parameters

Accurate model parameters are critical to the performance of the offline simulation and online observer. Below are all the major parameters that we considered.

TABLE I
MODEL PARAMETERS, FINAL VALUES, AND SOURCES

Parameters	Value	Source
Limb		
Upper limb length (m)	0.1	Measured
Lower limb length (m)	0.2	Measured
Upper limb mass (kg)	0.040	Measured
Lower limb mass (kg)	0.081	Measured
Upper limb inertia (kg-m^2)	0.00023	Estimated
Lower limb inertia (kg-m^2)	0.00068	Estimated
Motor		
Torque Constant (Nm/A)	0.0959	Estimated
Rotor Inertia (kg-m^2)	0.00005	Estimated
Static Friction (Nm)	0.056	[9]
Kinetic Friction (Nm)	0.023	[9]
Viscous Friction ($\frac{Nm}{rad/s}$)	0.00013	[9]
Armature Resistance (Ω)	0.180	Estimated
Others		
Joint Viscous Friction ($\frac{Nm}{rad/s}$)	0.0037	Estimated
Time delay (ms)	3	Estimated

1) Limbs: Limb length and mass are measured directly. Initially we estimated the limb inertia from the CAD model, but the extra weight of the bolts and bearings at the joints were ignored in that case.

2) Motors: The motor friction values are from [9], which also provided an overestimated value of the motor rotor inertia. The torque constant and armature resistance values are both provided by the motor manufacturer, but any partial burnout of the motors may have affected the values.

3) Time delay: For the offline model, the time delay between the mainboard and the motor controller needs to be taken into account. The effect of the time delay on the stability of the Minitaur robot is documented in [15]. When we ran simulation with a fairly long time delay (10ms), the result showed instability of the legs.

Overall, we estimated the limb inertia, torque constant, armature resistance, rotor inertia of the motor, and time delay. We collected the sample data by running the triangular trajectory of the legs in the free air, and the speed ramped up from 1 to 2.5 strides per second. The cost function was set to the total error per millisecond (per update) from all the samples. We chose the constrained Nelder-Mead method [16], which uses variable transformation to set the upper and lower bounds of the estimated parameters. We ran the method with different initial values within the bounds. The results were consistent and reasonable, listed in the Table 1. The estimated limb inertia is higher than the estimations from the CAD model, possibly due to the extra weight of bolts and bearings. The torque constant and armature resistance provided in the motor specifications is 0.0959 Nm/A and 0.186 Ω respectively, exactly the same as or very close to the estimation results. The estimated time delay value is consistent with [15].

C. Offline Simulation

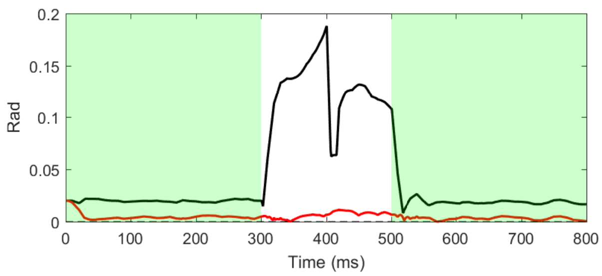
When the robot trots on the ground, the legs follow a pre-defined reference trajectory, an isosceles triangle in the Cartesian space. We simulated the trotting gait of the robot with 1ms time step in Matlab by solving the leg dynamics. The outputs of the simulation, $\hat{\theta}$ and $\dot{\hat{\theta}}$, are compared to the leg position θ and velocity $\dot{\theta}$ recorded in real robot trials, and the observer residuals of the simulation are formed by taking the average of the absolute values of both motors:

$$r_\theta = \frac{sum(|\hat{\theta} - \theta|)}{2} \tag{6}$$

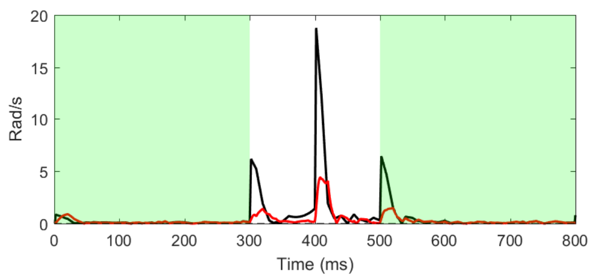
$$r_{\dot{\theta}} = \frac{sum(|\dot{\hat{\theta}} - \dot{\theta}|)}{2} \tag{7}$$

We collected leg data at a median speed (1.25 stride per second) in both conditions of swinging in the air and trotting on the ground, and plot r_θ and $r_{\dot{\theta}}$ with controller tracking error ($\theta_d - \theta$ and $\dot{\theta}_d - \dot{\theta}$) in Fig. 3 and 4. The shaded green region indicates the expected stance phase of the gait.

The figures show that the tracking errors in the expected flight phase is much higher than the observer residuals. Due to the nature of PD controller, the leg cannot follow the corners of the triangular trajectory well where high acceleration occurs, while the model-based simulation accounts for the inertial effect. Therefore, if the leg experiences any disturbance in the air, the increasing residual values should indicate the event,

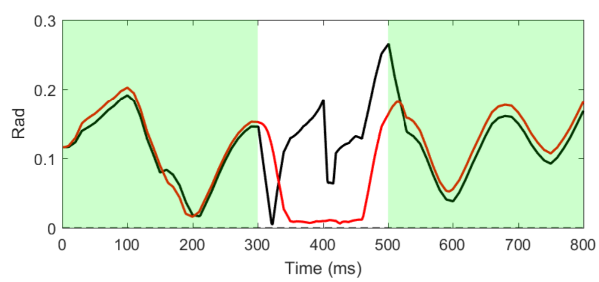


(a) Position residual (red), r_θ, and position tracking error (black), $\theta_d - \theta$

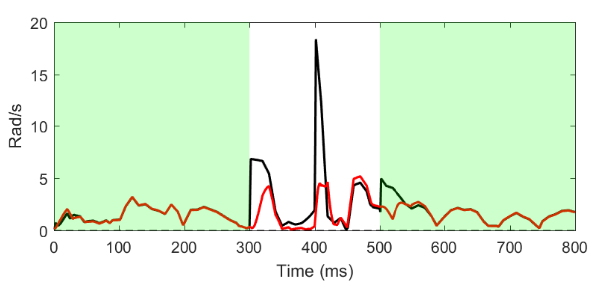


(b) Velocity residual (red), $r_{\dot{\theta}}$, and velocity tracking error (black), $\dot{\theta}_d - \dot{\theta}$

Fig. 3. Observer residual compared to PD controller tracking error in position and velocity in a free swinging cycle (no disturbance from the ground). Green shading indicates the expected stance phase.



(a) Position residual (red), r_θ, and position tracking error (black), $\theta_d - \theta$



(b) Velocity residual (red), $r_{\dot{\theta}}$, and velocity tracking error (black), $\dot{\theta}_d - \dot{\theta}$

Fig. 4. Observer residual compared to PD controller tracking error in position and velocity in a trotting stride cycle (disturbance from the ground). Green shading indicates the expected stance phase.

while the tracking errors would not be able to. Although the velocity residual values show the similar trend, the position data alone was sufficient to capture the abrupt change in leg dynamics.

Note that during the expected stance phase, both residual and tracking error values are high since we ignore the ground contact force in the leg dynamics. The leg disturbance during the stance phase is not within the interest of this paper, and would require contact force estimation.

D. Online Observer

While the robot trots on the flat ground with a pre-defined trajectory, the legs do not follow a specific trajectory when the robot climbs up stairs. Many of the dynamic maneuvers on the stairs are closely sequenced and some involve open-loop power. While this can still be simulated, the errors in residual values are likely to accumulate, and affect the initial conditions of the legs at a certain step within the sequenced behavior. This leads to the necessity of a memoryless, online observer that predicts leg states as the robot bounds up stairs. At each time step, the observer reads the current states of the legs from the motor encoders, and outputs the predicted states at the next time step for comparison.

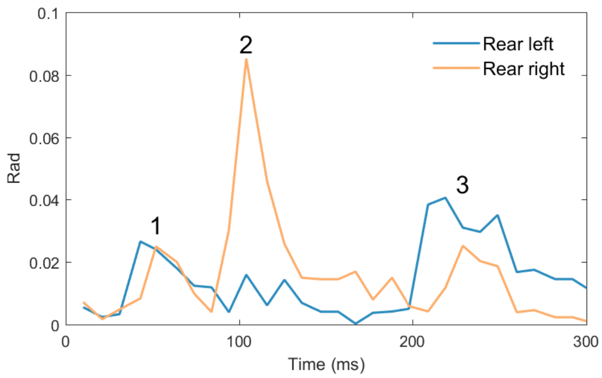
1) Reduced dynamics: Since the computational power of the onboard microcontroller is limited, we adopted a reduced dynamics model instead. The Coriolis terms take a fair amount of overall computations, but the values are usually small enough to be negligible. We also remove the joint friction terms. The equation of motion is reduced to:

$$M(\theta)\ddot{\theta} + N(\theta) = \tau \tag{8}$$

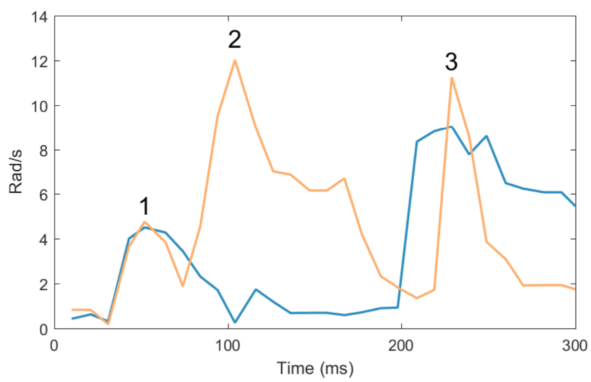
Since the robot always has at most two legs in the air when it climbs up stairs, only two observers run at the same time. The computations benchmark at about 2ms per cycle. However, the leg states do not change significantly in 2ms even if the leg hits an obstacle. When the rear legs lift off the ground and swing forward in the air, the residual shows a small amount of error due to the inertial effect, which is comparable to the residual caused by the obstacle. Therefore, we extended the observer time step from 2ms to 5ms, and the results were able to differentiate hitting the obstacle from swinging forward as shown in Fig. 5. Now the motor commands update at 200 Hz during stair ascent as opposed to 1 KHz during trotting. The update rate is still fast enough for quick reactive behaviors in the case of leg disturbances.

Given the changes in the model, we re-ran the parameter estimation using the same constrained Nelder-Mead method on the same selected parameters, except for the removed joint viscous friction coefficient. The results were very close to those listed in Table 1 and the differences are negligible.

To compare the performance of the reduced and full models, we calculated the position residuals summed in a free swinging cycle at a median speed (1.25 strides per second) using both models. We also simulated the 200 Hz update rate on both models. The results are listed in Table 2 below. The lower update rate does not affect the performance significantly. The model reduction results in an increase in errors, but still within a reasonable range. The goal of the online observer is to capture the sudden impulse at the leg instead of tracking a specific trajectory, thus not as demanding in performance. Fig. 5 shows that the reduced model is sufficient to pick up such impulses from obstacle contacts.



(a) Position residual



(b) Velocity residual

Fig. 5. Online observer residuals when the rear legs are in the air and the rear right leg hits an obstacle. The three peaks corresponds to: 1. swinging forward, 2. obstacle contact, 3. touchdown

TABLE II
PERFORMANCE OF THE FULL AND REDUCED MODELS AT DIFFERENT UPDATE RATED

Model Type	Update Rate (Hz)	Error (Rad)
Full	1000	7.57
Full	200	7.68
Reduced	1000	16.60
Reduced	200	16.86

III. Stair Climbing and Reactive Behavior

After we developed both the offline model-based simulation and the online observer, we would like to apply these strategies in rugged stair climbing. However, we quickly encountered difficulty in developing a reliable stair climbing gait for the Minitaur. Since the robot is not capable of walking up stairs, it has to use gaits that are more dynamic such as bounding. With no existing controller for stair ascent, tuning open-loop power and controller gains turned out to be very time-consuming. Another issue was that the supporting legs on the ground often slip off the stairs when the other pair bound, causing the robot to fall off the stairs. We reduced the slippage by applying open-loop power to generate more normal force. The current gait is robust enough to ensure that the robot does not slip off the stairs.

The stair climbing behavior consists of a few major components.

A. Trotting on the Floor and Stair Detection

The robot trots with a pre-defined triangular trajectory towards the first stair, and calculates the residual values using offline simulation outputs and motor encoder readings. The robot would check the residual values within a time range when the legs are expected in the air, determined based on offline simulation results.

Initially we would like to put some obstacles on the ground, and the robot would detect them and step over them with fast leg circulation. However, we could not find a method that consistently differentiates the obstacle and the stairs. Therefore the robot now treats the first stair as the disturbance and then starts stair climbing.

B. Legs Bounding

In order to climb the stairs, the front and rear legs of the robot bound upstairs alternatively. The bounding legs push off the ground, retract until minimum length, swing forward in the air, and touch down (Fig. 6). The supporting legs apply open-loop power into stair to generate normal force for balance. Meanwhile, two observers update the residual values of the bounding legs. While the leg touchdown generates high residual values, the expected touchdown time is known to the robot. The robot would treat the high position and velocity residual values as hitting an obstacle if it occurs before the expected touchdown.

C. Disturbance Recovery

Once the leg observer reports the presence of an obstacle, the affected leg retracts to the minimum length and swings forward, stepping onto the obstacle. Meanwhile, the two legs on the other diagonal extend more to lift up the whole robot body.

D. Trot on the stair tread

When both the front and rear legs have bounded up, there is usually some space between the front legs and the next stair. The front legs have to start bounding right next to the stair, otherwise they will not be able to catch it due to geometric limit. We rotate the triangular trajectory of the ground trotting to match the slope of the stairs, and the robot was able to trot forward with small steps and align itself with the next stair. After a certain time threshold, the next bounding cycle starts.

E. Exit the stairs

Detecting the end of staircase is straightforward. Once the IMU reports a very small pitch angle after a bounding cycle, the robot knows that it has reached the top of the stairs. It would trot forward for some distance and stop.

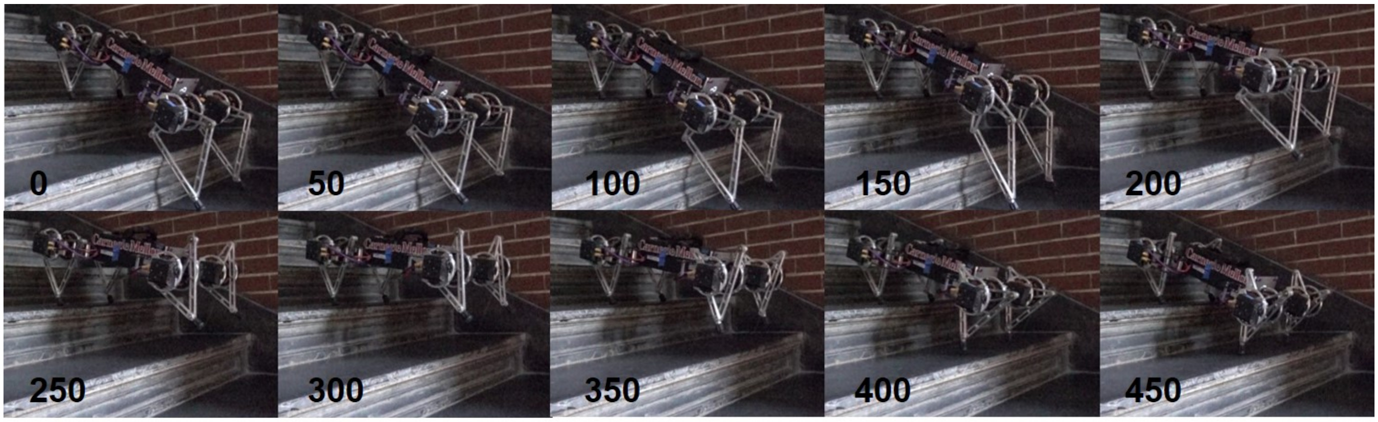


Fig. 6. The rear legs bound up the stair in 450ms.

IV. Experimental Results

We test the performance of the offline simulation, online observer, and stair-climbing gait separately. Eventually we would like to place random obstacles on the stairs and conduct an experiment survey on stairs of various dimensions and surface conditions.

1) Stair Detection with Offline Simulation: We let the robot start trotting at ten locations, ranging from 0.2 meter to 2 meter from the stairs. The robot was able to detect the stairs every time within two leg cycles once it contacted the stair.

Ideally the robot should detect the stair within the same leg cycle. However, we found that the stair contact sometimes occurred right before the leg hit the ground. If the end of the detection time range is set too close to the start of the ground phase, the robot often mistook the ground for the stair. Some delay on the ground is sacrificed for the extra accuracy of stair detection.

2) Obstacle Detection with Online Observer: A brick-like obstacle was placed on the stair, and tripped the rear right leg of the robot as it swung forward in the air. During initial tests, the robot was able to detect the obstacle consistently.

3) Stair Climbing Robustness: We challenged the robot to keep climbing the stairs before it failed. During initial tests, the robot was able to climb about five to ten stairs.

We realized that PWM control for bounding is not sufficient for robustness. As the robot climbed upstairs, the bounding height of the legs decreased given the same PWM values since the motors started to overheat and the battery charge is reduced. The bounding discrepancies caused the robot to trip on the stairs, or steer sideways and lose balance.

V. Conclusion

This work addresses the robustness of the stair climbing of a quadruped by incorporating proprioceptive sensing and climbing behavior. Offline simulation models and online observers are developed for disturbance detection. The robot is able to detect the stairs, bound up the stairs, and react to obstacles by quickly adjusting the leg trajectory. More experiments are yet to be completed. We believe that proprioceptive contact detection and reactive behavior would be crucial to the robust traversal of stairs.

Certainly the overall stair climbing framework is not quite complete yet. Parts of the climbing behavior we introduced rely on open-loop control, causing issues such as slipping of the supporting legs. Although our main focus has been using proprioceptive sensing to improve the robustness, we hope to establish a better stair climbing model for behavioral development, integrating force and impedance control. Another ongoing project at our lab is to investigate the effect of tails on dynamic maneuverability of the Minitaur robot. We believe that active actuation of an inertial or aerodynamic tail would maintain the stability of the robot during stair ascent, and even help achieve certain dynamic behavior that is otherwise infeasible.

Other future works include adding an extra onboard computer to enable online, full dynamics disturbance detection. The extra computational power also enables many other improvements, including whole body modeling, contact force sensing, sensor fusion, and sequential compositions [17] of online controllers. A newer version of the Minitaur robot also provides current sensing, which would allow motor current control and improve the accuracy of the online observer by skirting the ideal motor model.

Beyond the scope of stair climbing, we envision that proprioceptive sensing and model-based behavioral development would be critical components of the overall terrain locomotive framework of the legged robot. The robot may combine these components with other approaches such as vision detection to fulfill robust traversal through any terrains.

Acknowledgment

This work was supported by Robotics Institute Summer Scholar program and Robomechanics Lab at Carnegie Mellon University, PA. We would like to thank Joseph Norby, Nathan Kong, and Barrett Werner for the assistance and helpful discussions.

References

[1] S. Kim and P. Wensing, "Design of dynamic legged robots," vol. 5, pp. 117–190, 01 2017.

[2] T. T. Topping, G. Kenneally, and D. E. Koditschek, "Quasi-static and dynamic mismatch for door opening and stair climbing with a legged robot," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1080–1087.

[3] Y. Xiong and L. Matthies, "Vision-guided autonomous stair climbing," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, April 2000, pp. 1842–1847 vol.2.

[4] S. Steplight, G. Egnal, S. . Jung, D. B. Walker, C. J. Taylor, and J. P. Ostrowski, "A mode-based sensor fusion approach to robotic stair-climbing," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, vol. 2, Oct 2000, pp. 1113–1118 vol.2.

[5] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, D. M. Helmick, and L. Matthies, "Autonomous stair climbing for tracked vehicles," *The International Journal of Robotics Research*, vol. 26, no. 7, pp. 737–758, 2007.

[6] E. Z. Moore, D. Campbell, F. Grimminger, and M. Buehler, "Reliable stair climbing in the simple hexapod 'rhex'," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 3, May 2002, pp. 2222–2227 vol.3.

[7] G. J. Wenger, A. M. Johnson, C. J. Taylor, and D. E. Koditschek, "Semi-autonomous exploration of multi-floor buildings with a legged robot," in *Unmanned Systems Technology XVII*, vol. 9468. Baltimore, MD: SPIE, April 2015, pp. 94 680B–8.

[8] A. M. Johnson, M. T. Hale, G. C. Haynes, and D. E. Koditschek, "Autonomous legged hill and stairwell ascent," in *Proceedings of the IEEE Intl. Workshop on Safety, Security, & Rescue Robotics*, Kyoto, Japan, November 2011, pp. 134–142.

[9] G. Kenneally, A. De, and D. E. Koditschek, "Design principles for a family of direct-drive legged robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 900–907, July 2016.

[10] A. De and D. E. Koditschek, "Vertical hopper compositions for preflexive and feedback-stabilized quadrupedal bounding, pacing, pronking, and trotting," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 743–778, 2018.

[11] A. M. Johnson, G. C. Haynes, and D. E. Koditschek, "Disturbance detection, identification, and recovery by gait transition in legged robots," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 5347–5353.

[12] K. Kaneko, F. Kanehiro, M. Morisawa, E. Yoshida, and J. Laumond, "Disturbance observer that estimates external force acting on humanoid robots," in *2012 12th IEEE International Workshop on Advanced Motion Control (AMC)*, March 2012, pp. 1–6.

[13] L. Manuelli and R. Tedrake, "Localizing external contact using proprioceptive sensors: The contact particle filter," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 5062–5069.

[14] G. Bledt, P. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, to appear.

[15] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-Real: Learning Agile Locomotion For Quadruped Robots," *ArXiv e-prints*, Apr. 2018.

[16] J. D'Errico, "fminsearchbnd, fminsearchcon - file exchange - matlab central." [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/8277-fminsearchbnd-fminsearchcon

[17] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *The International Journal of Robotics Research*, vol. 18, no. 6, pp. 534–555, 1999.