

Stair Climbing via Successive Perching

Nicholas Morozovsky, *Member, IEEE*, and Thomas Bewley

Abstract—Stairs are a primary challenge for mobile robots navigating indoor human environments. Stair climbing is a useful, if not necessary, capability for mobile robots in urban search and rescue, security, cleaning, telepresence, elder care, and other applications. Existing stair climbing robots are large, expensive, and not always reliable, especially when descending stairs. In this paper, we present a novel approach for stair climbing that is achievable by a small mobile robot with minimal actuators and sensors and, thus, cost. The proposed robot has articulated tread assemblies on either side of a chassis. Using feedback control, the robot can balance on the edge of a single step. As the robot drives up the step, the chassis pivots to maintain the center of mass directly above the contact point. The dynamics of the system are derived with the Lagrangian method, and a discrete-time integral controller with friction compensation is designed to stabilize a stair climbing trajectory. The algorithms used to estimate the state of the system with low-cost noisy proprioceptive sensors are explained in detail. No external motion capture system is used. Simulation results are compared with successful experimental results.

Index Terms—Control systems, motion control, robot dynamics.

I. INTRODUCTION

IN order for robots to be accepted and useful in indoor human environments, they must be able to locomote unassisted. Three primary locomotion challenges, beyond stationary and moving obstacle avoidance, are stairs, doors, and thresholds (a degenerate case of stairs). In this paper, we focus on the stair climbing problem. A number of existing robots are capable of stair climbing, which can generally be classified into a few categories. These include humanoids, such as those featured in the DARPA Robotics Challenge [1]–[3]. Despite recent advances prompted by the challenge, it is not necessary, and indeed it is complex and costly, to locomote in a human manner in a human environment. Alternative form factors and control strategies can be simpler, cheaper, and faster. Traditional treaded vehicles [4], [5] that are long enough to span multiple step edges can climb

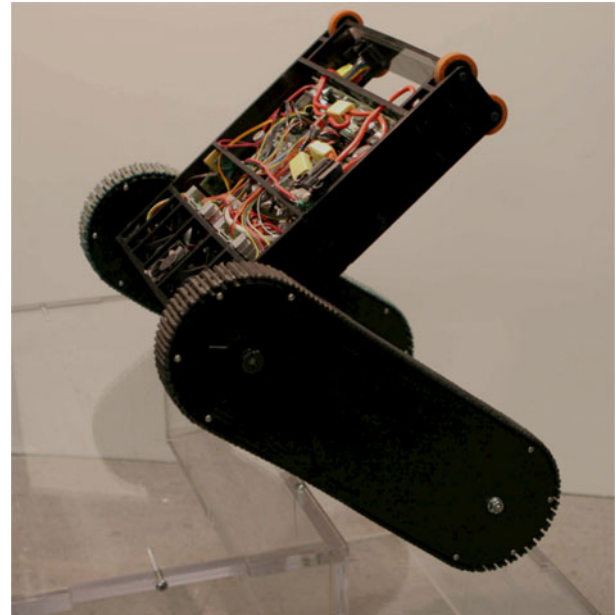


Fig. 1. Switchblade robot perching on the edge of a step.

stairs in a straightforward manner. Additional, articulated tread segments can aid in agility, particularly on the first and last steps. Another class of stair climbing robots utilize hybrid wheel-leg, or whleg, systems. This includes the popular RHex hexapod [6] and a number of robots with alternate whleg designs, including the DEKA iBOT [7] and others [8], [9]. A unique design utilizes deformable wheels which can transform into treads to fit into tight places as well as pivot-like whlegs to overcome steps, but the prototype presented is still limited to small stairs [15]. Other robots employ a dedicated mechanism for stair climbing, such as hopping [10], [11] or a system for raising and lowering the robot in two or more segments. While these methods can be effective at climbing stairs, the weight and size of the dedicated stair climbing components detract from the robot's performance on flat ground and adds to cost. There has been recent research into stair climbing robots with mobile inverted pendulum dynamics [13], [14], but both are wheeled systems with wheel radii greater than the step height. Another wheeled inverted pendulum robot is smaller and uses a novel mechanism to climb its own central post, but requires space on each step to reposition itself and is slow [12]. A number of stair climbing robots are not capable of climbing standard stairs, that is, they can only climb stairs with an unrealistically low pitch angle, and/or rise. Some of the vehicles would be difficult, costly, and inefficient to scale to the size of standard stairs.

The proposed method of stair climbing via successive perching combines dynamic inverted pendulum balancing with the mechanical simplicity of a treaded design. The robot approaches the first step balanced on the far end of the tread assemblies with the chassis angled back to keep the center of mass above

Manuscript received October 14, 2014; revised January 13, 2015 and April 8, 2015; accepted April 17, 2015. Date of publication April 27, 2015; date of current version October 21, 2015. Recommended by Technical Editor S. C. Mukhopadhyay. This paper was presented in part at the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011.

The authors are with the Coordinated Robotics Lab, University of California, San Diego, CA 92093 USA (e-mail: nmorozov@ucsd.edu; bewley@ucsd.edu).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org> provided by the authors. The material is 40.2 MB. This video shows hardware tests and an animation of the Switchblade robot, developed at the University of California San Diego Coordinated Robotics Lab. The robot is shown performing several maneuvers, including the perching maneuver discussed in the accompanying paper. Results from stabilizing the horizontal equilibrium position while perching demonstrate the benefit of the friction compensator. The robot is also shown traversing a distance of 10 cm at an inclination angle of 10 degrees in 10 seconds. The video can be viewed with QuickTime 7 Player, available for Mac and Windows. QuickTime can be downloaded for free at: <http://www.apple.com/quicktime/download/>. Contact nmorozov@ucsd.edu for further questions about this work.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMECH.2015.2426722

the contact point with the ground. Once the tread assemblies make contact with the first step edge, the chassis pivots forward, shifting the center of mass directly above the step edge. The robot then drives up the step edge, pivoting the chassis appropriately to maintain the center of mass above the contact point and balancing dynamically (see Fig. 1). At the top of the step, the robot transitions to balancing on the top face of the step and can then climb successive stairs similarly. See the included video for an animation of the stair climbing sequence. This maneuver is compatible with a wide range of step dimensions. This paradigm requires only that the length of the tread, not the sprocket radius, be greater than the rise of the step. Further, the length of the tread does not need to span multiple step edges, as in other treaded stair climbing robots. The smaller required size of the robot decreases cost as well as increases maneuverability in tight spaces, such as in a partially collapsed building. No additional or dedicated sensors or actuators are required for stair climbing, and no external feedback system, such as a vision system, is required or used.

The (patent pending) robot, Switchblade, used to test this method of stair climbing has been developed at the University of California, San Diego Coordinated Robotics Lab [16]. As in a traditional treaded vehicle, Switchblade has a pair of tread assemblies, driven by an internal sprocket, mounted on either side of a central chassis. Uniquely, the tread assemblies can rotate continuously about the main drive axle of the chassis. Changing the angle between the chassis and tread assemblies moves the center of mass. There are no physical connections between the left and right tread assemblies to keep them parallel, but feedback control may be applied when it is desired to keep the two tread assemblies in line.

In a horizontal configuration, the robot functions much like any other treaded skid-steer robot with the ability to independently drive each tread forward or backward to drive and turn. The treads act to minimize contact force on loose surfaces and maintain traction better than wheels. Note that the actuated tread assemblies make the robot impervious to high centering. Note also that the robot operates just as effectively “upside down” as “right side up.”

We have previously described a rudimentary stair climbing maneuver that can be performed with the same vehicle. Climbing successive stairs required each step to have sufficient run for the robot to be able to turn around or flip itself. This method is still effective for thresholds such as street curbs. The maneuver presented in this paper, dubbed successive perching, does not have this limitation on step geometry. An overview of the maneuvers the vehicle is capable of is shown in the included video. This vehicle design has been recognized for scoring well in both versatility and mechanical complexity metrics [17].

In the following sections, we will present the nonlinear dynamics of the perching problem and then describe the controller we use to stabilize a stair climbing trajectory. We next discuss the physical prototype robot used in the experiments and the estimation technique. We close with an analysis of experimental results and a discussion of future work.

II. DYNAMICS

In order to create a simulation and design a controller, we first derive the dynamics of the system using the Lagrangian

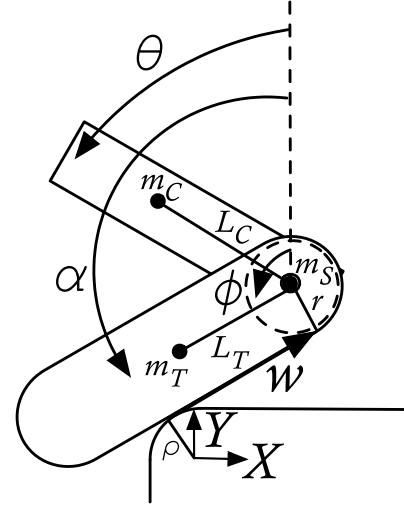


Fig. 2. Generalized coordinates and key dimensions.

method. By symmetry, we simplify the model to three bodies in two dimensions, the generalized coordinates are defined in Fig. 2, θ and α are the angles of the chassis and unified right and left tread assemblies from vertical, respectively, ϕ is the rotation angle of the unified right and left tread sprockets from vertical, and w is the distance between the tread sprocket and the contact point on the step edge measured along the tread. The radius of curvature of the step edge is ρ ; a sharp edge is modeled by setting $\rho = 0$. The inertial frame XY is fixed at the center of the curved edge. Motors on the robot exert torques between the tread sprocket and the chassis and between the chassis and the tread assembly.

The kinetic energies of the unified sprockets, unified tread assemblies, chassis, unified sprocket motors, and unified tread assembly motors are given, respectively, by

$$\begin{aligned}
 T_S &= \frac{1}{2} \{ m_S [(\dot{w} \cos \alpha - \dot{\alpha}(r \cos \alpha + \rho \cos \alpha + w \sin \alpha))^2 \\
 &\quad + (\dot{w} \sin \alpha - \dot{\alpha}(r \sin \alpha - w \cos \alpha + \rho \sin \alpha))^2] + J_S \dot{\phi}^2 \} \\
 T_T &= \frac{1}{2} \{ m_T [(\dot{\alpha}(r \cos \alpha - L_T \sin \alpha + \rho \cos \alpha + w \sin \alpha) \\
 &\quad - \dot{w} \cos \alpha)^2 + (\dot{\alpha}(L_T \cos \alpha + r \sin \alpha - w \cos \alpha \\
 &\quad + \rho \sin \alpha) - \dot{w} \sin \alpha)^2] + J_T \dot{\alpha}^2 \} \\
 T_C &= \frac{1}{2} \{ m_C [(\dot{w} \cos \alpha - \dot{\alpha}(r \cos \alpha + \rho \cos \alpha + w \sin \alpha) \\
 &\quad + L_C \dot{\theta} \sin \theta)^2 + (\dot{\alpha}(r \sin \alpha - w \cos \alpha + \rho \sin \alpha) \\
 &\quad - \dot{w} \sin \alpha + L_C \dot{\theta} \cos \theta)^2] + J_C \dot{\theta}^2 \} \\
 T_{SM} &= \frac{1}{2} [J_{SM} (\dot{\phi} - \dot{\theta})^2], \quad T_{TM} = \frac{1}{2} [J_{TM} (\dot{\alpha} - \dot{\theta})^2]
 \end{aligned}$$

where m_S and J_S are the mass and rotational inertia, respectively, of the combined right and left tread sprockets, m_T and J_T are the mass and rotational inertia, respectively, of the combined right and left tread assemblies, L_T is the distance between the center of mass of the tread assemblies and the axis of rotation with the chassis, m_C and J_C are the mass and rotational inertia of the chassis, respectively, and L_C is the dis-

tance between the center of mass of the chassis and the axis of rotation with the tread assemblies. T_{SM} and T_{TM} are the kinetic energies of the combined right and left sprocket and tread assembly motors, respectively. The energy is only nonzero when there is relative rotation between either the sprockets and the chassis or the tread assemblies and the chassis. The motor coils do not rotate (and, therefore, do not have rotational kinetic energy when there is no relative rotation. The kinetic energy of the mass of the motors is accounted for in the kinetic energy of the chassis. J_{SM} and J_{TM} are the effective rotational inertias of the combined right and left sprocket and tread assembly motor coils, respectively. Motor coil rotational inertia is nominally small, but when a gear reduction is used, the inertia is effectively multiplied by the square of the gear ratio because it is rotating at a much higher velocity and the inertia can become significant. The gravitational potential energy is given by

$$V = g[m_S(r \sin \alpha - w \cos \alpha + \rho \sin \alpha) + m_T(L_T \cos \alpha + r \sin \alpha - w \cos \alpha + \rho \sin \alpha) + m_C(L_C \cos \theta + r \sin \alpha - w \cos \alpha + \rho \sin \alpha)].$$

We define the generalized coordinates as

$$q = (w \ \phi \ \alpha \ \theta)^T.$$

The Lagrangian can be written as $\mathcal{L} = T_S + T_T + T_C + T_{SM} + T_{TM} - V$. By solving the Euler-Lagrange equations, we can write the equations of motion in the form

$$M(q)\ddot{q} + F(q, \dot{q}) = B\tau + \frac{\delta P(\dot{q})}{\delta \dot{q}}. \quad (1)$$

The (positive definite) mass matrix $M(q)$ is given by:

$$\begin{aligned} M_{1,1}(q) &= m, & M_{1,2}(q) &= 0 \\ M_{1,3}(q) &= -(r + \rho)m, & M_{1,4}(q) &= -L_C m_C \sin(\alpha - \theta) \\ M_{2,2}(q) &= J_S + J_{SM}, & M_{2,3}(q) &= 0 \\ M_{2,4}(q) &= -J_{SM} \\ M_{3,3}(q) &= J_T + J_{TM} + m_T L_T (L_T - 2w) \\ &\quad + m[(r + \rho)^2 + w^2] \\ M_{3,4}(q) &= L_C m_C [(r + \rho) \sin(\alpha - \theta) - w \cos(\alpha - \theta)] \\ &\quad - J_{TM} \\ M_{4,4}(q) &= m_C L_C^2 + J_C + J_{SM} + J_{TM} \end{aligned}$$

where $m = m_S + m_T + m_C$. The vector $F(q, \dot{q})$ is given by

$$\begin{aligned} F_1(q, \dot{q}) &= m_T L_T \dot{\alpha}^2 - m w \dot{\alpha}^2 - m g \cos \alpha \\ &\quad + m_C L_C \dot{\theta}^2 \cos(\alpha - \theta) \\ F_2(q, \dot{q}) &= 0 \\ F_3(q, \dot{q}) &= 2\dot{\alpha} m w \dot{w} + m g (r + \rho) \cos \alpha \\ &\quad + g (m w - m_T L_T) \sin \alpha - 2L_T \dot{\alpha} m_T \dot{w} \\ &\quad - m_C L_C \dot{\theta}^2 [(r + \rho) \cos(\alpha - \theta) + w \sin(\alpha - \theta)] \\ F_4(q, \dot{q}) &= L_C m_C [(r + \rho) \dot{\alpha}^2 \cos(\alpha - \theta) \\ &\quad - 2\dot{\alpha} \dot{w} \cos(\alpha - \theta) + \dot{\alpha}^2 w \sin(\alpha - \theta) - g \sin \theta]. \end{aligned}$$

The right-hand side of (1) is the sum of the generalized forces of the system. The power dissipation function $P(\dot{q})$ [18] accounts for the Coulomb friction of the treads rubbing against the tread assemblies and the Coulomb friction between the chassis and tread assemblies

$$P(\dot{q}) = -\frac{\mu_k m g r}{\sin \alpha} (\dot{\phi} - \dot{\alpha}) - c_T (\dot{\alpha} - \dot{\theta}) \quad (2)$$

where μ_k is the coefficient of kinetic friction between the treads and tread assemblies and $m g / \sin \alpha$ is the normal force acting on the treads from the step edge at equilibrium. The coefficient c_T is a constant defined by the physical parameters of the system (mass, length, coefficient of kinetic friction between the chassis and tread assemblies, and gravitational acceleration). The contribution to the generalized forces can be determined from $F_q = \delta P(\dot{q}) / \delta \dot{q}$, where the signum function is appended to capture the direction-dependent nature of Coulomb friction

$$\frac{\delta P(\dot{q})}{\delta \dot{q}} = \begin{pmatrix} 0 \\ -\frac{\mu_k m g r}{\sin \alpha} \cdot \text{sgn}(\dot{\phi} - \dot{\alpha}) \\ \frac{\mu_k m g r}{\sin \alpha} \cdot \text{sgn}(\dot{\phi} - \dot{\alpha}) - c_T \cdot \text{sgn}(\dot{\alpha} - \dot{\theta}) \\ c_T \cdot \text{sgn}(\dot{\alpha} - \dot{\theta}) \end{pmatrix}. \quad (3)$$

Alternatively, a smooth model, such as used in [19], could be used to model the friction. The matrix B in (1) maps τ , the control input torque vector for the motors in the chassis, to the generalized coordinates

$$B = \begin{bmatrix} 0 & 0 \\ -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix}. \quad (4)$$

The first element of τ represents the motor torque between the chassis and the tread sprockets and the second element represents the motor torque between the chassis and the tread assemblies. The motors are electrically connected to exert a positive torque on the chassis and equal and opposite reaction torques on the sprockets and tread assemblies when a positive voltage is applied, thus the paired positive and negative elements in (4) above. We model the torque output τ_κ from each motor linearly as

$$\tau_\kappa = \sigma_\kappa u_\kappa - \zeta_\kappa \omega_\kappa, \quad \sigma_\kappa = \frac{\gamma_\kappa k_\kappa V}{R_\kappa}, \quad \zeta_\kappa = \frac{(\gamma_\kappa k_\kappa)^2}{R_\kappa} \quad (5)$$

where σ_κ is the stall torque, u_κ is the control input (limited to $[-1, 1]$), ζ_κ is the back EMF damping coefficient of the motor, ω_κ is the speed of the motor shaft relative to the motor body, k_κ is the motor constant, γ_κ is the gear ratio of the transmission, V is the nominal battery voltage, and R_κ is the terminal resistance [20]. Substituting the motor model (5), we can rewrite (1)

$$M(q)\ddot{q} + F(q, \dot{q}) = B[\Sigma u - Z(\dot{q})] + \frac{\delta P(\dot{q})}{\delta \dot{q}} \quad (6)$$

$$\Sigma = \begin{bmatrix} \sigma_S & 0 \\ 0 & \sigma_T \end{bmatrix}, \quad Z(\dot{q}) = \begin{pmatrix} \zeta_S(\dot{\theta} - \dot{\phi}) \\ \zeta_T(\dot{\theta} - \dot{\alpha}) \end{pmatrix}.$$

We next impose a no-slip constraint between the tread sprocket and the step edge, which will be shown also achieves a coordinate reduction, via

$$w + r(\phi - \alpha) + \rho(\pi/2 - \alpha) = 0. \quad (7)$$

In two dimensions, this is a holonomic constraint which we can differentiate with respect to time to write in the form

$$A_1 \dot{q} = 0, \quad A_1 = (1 \ r \ - (r + \rho) \ 0).$$

Alternately, we could explicitly solve (7) for w and replace w in all expressions, but this greatly expands the form of $M(q)$ and $F(q, \dot{q})$ and we lose the intuitive insight into which terms are dependent directly on the tread displacement and velocity. Also, it will be shown that by treating the no-slip constraint in this manner, we can apply it simultaneously with the nonholonomic constraints.

We can write additional nonholonomic constraints depending on whether the treads are in stiction with (not moving relative to) the tread assemblies ($\dot{\phi} = \dot{\alpha}$)

$$A_2 \dot{q} = 0, \quad A_2 = (0 \ 1 \ -1 \ 0)$$

or the tread assemblies are in stiction with the chassis ($\dot{\alpha} = \dot{\theta}$)

$$A_3 \dot{q} = 0, \quad A_3 = (0 \ 0 \ 1 \ -1).$$

These three constraints can be combined by stacking the row vectors to form a constraint matrix A_β . In this system, A_β is not dependent on q . We append (6) with the inner product of the constraint matrix A_β with λ_β , the Lagrange multiplier

$$M(q)\ddot{q} + F(q, \dot{q}) = B[\Sigma u - Z(\dot{q})] + \frac{\delta P(\dot{q})}{\delta \dot{q}} + A_\beta^T \lambda_\beta. \quad (8)$$

We assume that the no-slip constraint always holds, but are interested in the different combinations of tread/tread assembly and chassis/tread assembly stiction. This results in four possible constraint matrices:

- 1) no-slip only, $A_{NS} = A_1$;
- 2) no-slip with treads in stiction, $A_T = [A_1; A_2]$;
- 3) no-slip with chassis in stiction, $A_C = [A_1; A_3]$;
- 4) no-slip with treads and chassis in stiction, $A_{TC} = [A_1; A_2; A_3]$.

We can find orthonormal bases S_β for the null spaces of A_β

$$S_{NS} = \begin{bmatrix} -r/\sqrt{r^2+1} & (r+\rho)/\sqrt{(r+\rho)^2+1} & 0 \\ 1/\sqrt{r^2+1} & 0 & 0 \\ 0 & 1/\sqrt{(r+\rho)^2+1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$$S_T = \begin{bmatrix} \rho/\sqrt{\rho^2+2} & 0 \\ 1/\sqrt{\rho^2+2} & 0 \\ 1/\sqrt{\rho^2+2} & 0 \\ 0 & 1 \end{bmatrix}$$

$$S_C = \begin{bmatrix} -r/\sqrt{r^2+1} & (r+\rho)/\sqrt{(r+\rho)^2+2} \\ 1/\sqrt{r^2+1} & 0 \\ 0 & 1/\sqrt{(r+\rho)^2+2} \\ 0 & 1/\sqrt{(r+\rho)^2+2} \end{bmatrix}$$

$$S_{TC} = \begin{bmatrix} \rho/\sqrt{\rho^2+3} \\ 1/\sqrt{\rho^2+3} \\ 1/\sqrt{\rho^2+3} \\ 1/\sqrt{\rho^2+3} \end{bmatrix}.$$

Given that $\dot{q}$ is in this space, we define ν_β and $\dot{\nu}_\beta$ accordingly

$$\dot{q} = S_\beta \nu_\beta, \quad \ddot{q} = S_\beta \dot{\nu}_\beta \quad (10)$$

since S_β are constant-valued matrices. Premultiplying by S_β^T and using (10), we can rewrite (8) as

$$S_\beta^T M(q) S_\beta \dot{\nu}_\beta + S_\beta^T F(q, \dot{q}) = S_\beta^T B[\Sigma u - Z(\dot{q})] + S_\beta^T \frac{\delta P(\dot{q})}{\delta \dot{q}}.$$

Solving for the acceleration terms $\dot{\nu}_\beta$

$$\dot{\nu}_\beta = [S_\beta^T M(q) S_\beta]^{-1} S_\beta^T \{B[\Sigma u - Z(\dot{q})] + \frac{\delta P(\dot{q})}{\delta \dot{q}} - F(q, \dot{q})\}. \quad (11)$$

Premultiplying by S_β and using (10)

$$\ddot{q} = S_\beta [S_\beta^T M(q) S_\beta]^{-1} S_\beta^T \{B[\Sigma u - Z(\dot{q})] + \frac{\delta P(\dot{q})}{\delta \dot{q}} - F(q, \dot{q})\} \quad (12)$$

which is a set of second-order nonlinear differential equations which can be marched forward in time by traditional means, choosing the appropriate S_β as time progresses as a function of the state (see Section III-G).

III. CONTROLLER DESIGN

When the system is in stiction, the control authority is reduced. In the worst case, when both the treads and chassis are in stiction ($S_\beta = S_{TC}$), there is no effect of the motor torque on the treads or tread assemblies until enough torque is applied to break the stiction ($S_{TC}^T B = 0_{1 \times 2}$). We, therefore, focus our control design on the case where neither the treads nor the chassis are in stiction ($S_\beta = S_{NS}$). Noting that the dynamics of w are directly coupled to ϕ and α by (7), we can choose a reduced coordinate set $q_r = (\phi \ \alpha \ \theta)^T$ such that $\dot{q}_r = S'_{NS} \nu_{NS}$ where S'_{NS} is the bottom three rows of (9). Similarly, $\dot{q}_r = S'_{NS} \dot{\nu}_{NS}$.

Concatenating q_r and $\dot{q}_r$ yields a complete state vector x . Rewriting (11) and premultiplying by S'_{NS} to recover $\dot{q}_r$ from $\dot{\nu}_{NS}$, we see that this nonlinear system is affine in the inputs

$$x = \begin{pmatrix} q_r \\ \dot{q}_r \end{pmatrix}, \quad \dot{x} = f(x) + \Gamma(x)u$$

$$f(x) = \begin{pmatrix} \dot{q}_r \\ S'_{NS} [S_{NS}^T M(q) S_{NS}]^{-1} S_{NS}^T [\frac{\delta P(\dot{q})}{\delta \dot{q}} - BZ(\dot{q}) - F(q, \dot{q})] \end{pmatrix}$$

$$\Gamma(x) = \begin{bmatrix} 0_{3 \times 2} \\ S'_{NS} [S_{NS}^T M(q) S_{NS}]^{-1} S_{NS}^T B \Sigma \end{bmatrix}.$$

A. Equilibrium Conditions

We seek to find equations to describe the static equilibrium manifold of the system, that is, we seek to find expressions for x^* and u^* such that $\dot{x} = f(x^*) + \Gamma(x^*)u^* = 0_{6 \times 1}$ given $\dot{q}_r = 0_{3 \times 1}$.

Given that $S_{NS}^T M(q) S_{NS}$ is positive definite, we can solve the matrix expression

$$S_{NS}^T \{B[\Sigma u - Z(\dot{q})] + \frac{\delta P(\dot{q})}{\delta \dot{q}} - F(q, \dot{q})\} = 0_{3 \times 1} \quad (13)$$

from (11) and simplify by setting $\dot{q}_r = 0_{3 \times 1}$ to get three equations

$$\begin{aligned} -(\sigma_S u_1^* + mgr \cos \alpha^*) / \sqrt{r^2 + 1} &= 0 \\ -[\sigma_T u_2^* + g(mw^* - m_T L_T) \sin \alpha^*] / \sqrt{(r + \rho)^2 + 1} &= 0 \\ \sigma_S u_1^* + \sigma_T u_2^* + m_C L_C g \sin \theta^* &= 0. \end{aligned} \quad (14)$$

By inspecting the above equations, we see that there are unique solutions for the feedforward terms u_1^* and u_2^* in terms of ϕ^* and α^* , where $w^* = f(\phi^*, \alpha^*)$ (7)

$$u_1^* = -mgr \cos \alpha^* / \sigma_S \quad (15)$$

$$u_2^* = g(m_T L_T - mw^*) \sin \alpha^* / \sigma_T. \quad (16)$$

The expression for u_1^* is equivalent to the torque required to hold position at an inclination of α^* . The expression for u_2^* is the torque required to hold up the weight of the chassis from the tread assemblies. We also see that there is a unique solution for θ^* as a function of ϕ^* and α^* , combining (14)–(16)

$$\theta^* = \arcsin \left(\frac{mr \cos \alpha^* - (m_T L_T - mw^*) \sin \alpha^*}{m_C L_C} \right). \quad (17)$$

The equilibrium values ϕ^* , α^* , and θ^* correspond to a pose in which the overall center of mass is directly above the step edge. This can be confirmed by a static analysis of the system.

We define $\tilde{x} = x - x^*$ and $\tilde{u} = u - u^*$ such that

$$\dot{\tilde{x}} = f(\tilde{x} + x^*) + \Gamma(\tilde{x} + x^*)(\tilde{u} + u^*). \quad (18)$$

B. Friction Compensation

The signum function in (3) due to Coulomb friction cannot be linearized about the origin. Instead, we ignore this term in the linearization and add a separate friction compensator u_F to the controller [21]. The linearizable plant dynamics are

$$f_l(x) = \begin{pmatrix} -S'_{NS} [S'^T_{NS} M(q) S]^{-1} S'^T_{NS} [BZ(\dot{q}) + F(q, \dot{q})] \\ \dot{q}_r \end{pmatrix}. \quad (19)$$

There are a number of factors to be considered when designing the friction compensator. The compensator should mitigate both stiction and Coulomb friction without destabilizing the equilibrium manifold. An obvious choice to “eliminate” the Coulomb friction may be

$$u_F = \begin{pmatrix} -\frac{\mu_k m g T}{\sigma_S \sin \alpha} \cdot \text{sgn}(\dot{\phi} - \dot{\alpha}) \\ -(c_T / \sigma_T) \cdot \text{sgn}(\dot{\alpha} - \dot{\theta}) \end{pmatrix}.$$

However, practical matters such as backlash and chatter in the physical system limit the use of the signum function as a candidate compensator. We instead saturate a steep line passing through the origin (see Fig. 3). It is also inherently difficult to measure near-zero relative velocity with optical encoders (more in Section IV-C) and so smoother performance is possible when using the sign of $\tilde{u}_\psi$ instead of the relative velocity. We also desire a function that is simple to implement on an embedded controller (see Section IV-B). We come to a friction compensator with the form

$$u_F = \begin{pmatrix} \min(\max(\tilde{u}_1 a_S / b_S, -a_S), a_S) / \sin \alpha \\ \min(\max(\tilde{u}_2 a_T / b_T, -a_T), a_T) \end{pmatrix} \quad (20)$$

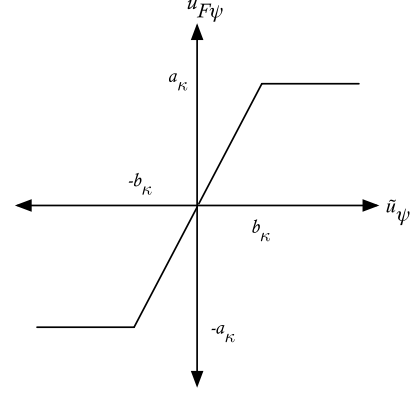


Fig. 3. Friction compensator $u_{F\psi}$ as a function of $\tilde{u}_\psi$.

which is illustrated in Fig. 3 and where $a_S \leq \mu_k mgr / \sigma_S$, $a_T \leq c_T / \sigma_T$, and a_κ , b_κ can be tuned empirically on the physical system. The effect of the friction compensator can be seen in Section IV-D.

C. Linearization and Integral Control

We linearize the system at the origin of the transformed system (18) using the linearizable plant dynamics (19)

$$\begin{aligned} \dot{\tilde{x}} &= A\tilde{x} + B(\tilde{u} + u^*), \\ A &= \left. \frac{\delta f_l(\tilde{x} + x^*)}{\delta \tilde{x}} \right|_{\tilde{x}=0}, \quad B = \Gamma(\tilde{x} + x^*) \Big|_{\tilde{x}=0}. \end{aligned}$$

By construction, the top half of A will have the form $[0_{3 \times 3} \ I_{3 \times 3}]$, and the top half of B is zeros because the system inputs affect the accelerations, not the velocities. In order to increase the robustness of the system to disturbances such as parameter and sensor error, we augment the state vector $\tilde{x}$ with the integrated regulation error ξ [22], defined by

$$\dot{\xi} = \begin{pmatrix} (\phi - \alpha) - (\phi^* - \alpha^*) \\ (\alpha - \theta) - (\alpha^* - \theta^*) \end{pmatrix}$$

noting that $(\phi - \alpha)$ is approximately w , the tread displacement, when $r \gg \rho$ and $(\alpha - \theta)$ is the separation angle between the chassis and the tread assemblies. Thus, the integrated regulation error will increase when the state is not at the desired tread displacement or chassis separation angle. We further define

$$\bar{x} = \begin{pmatrix} \tilde{x} \\ \xi \end{pmatrix}, \quad C = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \end{pmatrix}$$

and the system can now be written

$$\dot{\bar{x}} = \mathcal{A}\bar{x} + \mathcal{B}(\tilde{u} + u^*) \quad (21)$$

$$\mathcal{A} = \begin{bmatrix} A & 0_{6 \times 2} \\ C & 0_{2 \times 2} \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} B \\ 0_{2 \times 2} \end{bmatrix}.$$

D. Linear Quadratic Regulator (LQR)

A state feedback gain matrix can be found using the LQR method. The weighting matrices are determined by Bryson's method [23]. For the horizontal equilibrium where $\alpha^* = \theta^* =$

$$\pi/2$$

$$Q_C = \text{diag} \left(\frac{1}{(k)^2} \frac{1}{(k)^2} \frac{1}{(k)^2} \frac{1}{(\pi)^2} \frac{1}{(l)^2} \frac{1}{(l)^2} \frac{1}{(4/5)^2} \frac{1}{(7/5)^2} \right)$$

$$R_C = \text{diag} \left(\frac{1}{(2/5)^2} \frac{1}{(1/4)^2} \right), \quad N_C = 0_{7 \times 2}$$

where $k = 1/8$ and $l = 7/2$.

E. Discretization

Since the control will be implemented with digital electronics, we must discretize the system; we choose a sample time of $h = 0.01$ s. We convert our continuous-time system from (21) using the matrix exponential

$$\begin{aligned} \bar{x}_{k+1} &= F\bar{x}_k + G(\tilde{u}_k + u^*) \\ F &= \Phi(h), \quad \Phi(\tau) = e^{A\tau} \\ G &= \Theta(h), \quad \Theta(\tau) = \int_0^\tau e^{A\eta} B d\eta. \end{aligned}$$

The continuous-time weighting matrices are also transformed (given that $N_C = 0$)

$$\begin{aligned} Q_D &= \int_0^h \Phi^T(\tau) Q_C \Phi(\tau) d\tau \\ R_D &= \int_0^h \Theta^T(\tau) Q_C \Theta(\tau) + R_C d\tau \\ N_D &= \int_0^h \Phi^T(\tau) Q_C \Theta(\tau) d\tau. \end{aligned}$$

A discrete state feedback matrix K is found using the discrete-time LQR method $\tilde{u} = K\bar{x}$. The final control law is of the form $u = K\bar{x} + u^* + u_F$. As noted, the motor model (5) is valid for a bounded control input $\in [-1, 1]$, so each element of u is saturated at unity magnitude.

F. Trajectory Generation and Gain Scheduling

To drive up a step edge, we plan a trajectory along the static equilibrium manifold such that every point in the trajectory satisfies the equilibrium conditions (13). The overall center of mass is continuously maintained directly above the step edge. This approach has the added benefit that the trajectory can simply be reversed to descend a step edge while maintaining equilibrium. We vary w^* from zero to the length of the tread assembly and choose the tread inclination angle α^* to be constant, from which we can compute a range of values for $\phi^* \in [\phi_S, \phi_F]$ (7) which correspond to the start and final tread displacement positions. The value of θ^* is given by (17).

The dynamics (19) change considerably across this range as the angle of the chassis θ^* and, thus the height of center of mass, changes, so we choose $n_G = 5$ values of ϕ^* evenly distributed $\in [\phi_S, \phi_F]$, use our constant α^* , solve for θ^* , and find a discrete-time state feedback matrix K at each equilibrium point. The weighting matrices Q_C and R_C are adjusted at each equilibrium point to keep the norm of K constant. Five different positions along a step edge climbing equilibrium manifold are shown in Fig. 4. We construct an $8 \times 2 \times n_G$ lookup table containing all of the state feedback matrices (each 8×2) indexed by the

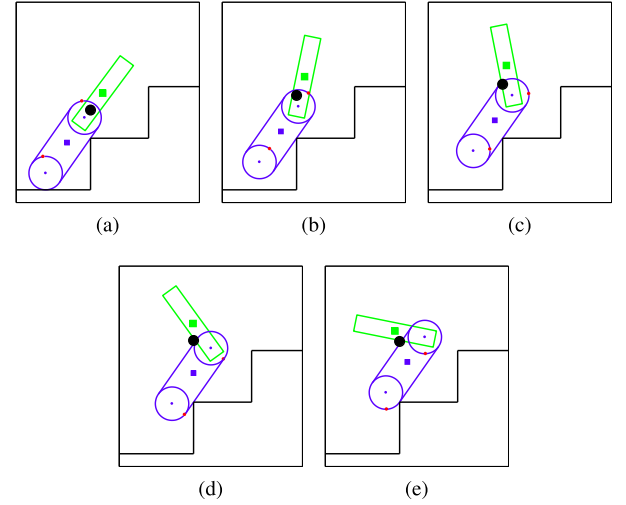


Fig. 4. Stair climbing trajectory with centers of mass of chassis and tread assemblies as squares, and overall center of mass as a circle.

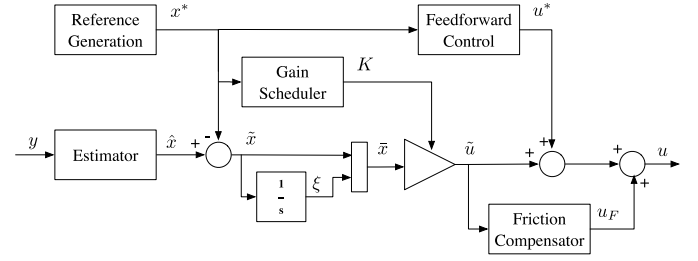


Fig. 5. Controller block diagram.

reference value of ϕ^* and implement gain scheduling by linearly interpolating between the entries of the lookup table depending on the current reference command ϕ^* . A block diagram of the controller is shown in Fig. 5. The estimator is described in Section IV-C.

G. Simulation

We created a simulation in MATLAB and Simulink in order to validate the model and controller. The full nonlinear continuous-time dynamics of the system (12) are used to represent the plant and are numerically integrated with a variable time step Runge-Kutta method [24]. The controller is implemented in discrete time as per Section III with a zero-order hold on the motor input u between control loops. The choice of which constraints S_β in the plant model to apply at time step t_i is determined by logical functions of the state vector $\bar{x}_i$, the control input u_i , and the previous constraint condition. Starting at rest, both the treads and the chassis are assumed to be in stiction, $S_\beta = S_{TC}$. If and when there is sufficient torque (from combined contributions of $F(q, \dot{q})$ and u) to break the stiction between the treads and the tread assemblies or the chassis and the tread assemblies, then a different S_β is used. If the relative velocity between the treads and the tread assemblies or the chassis and tread assemblies changes sign (passes through zero), then the system reenters stiction. Entering stiction is treated as an inelastic collision, where the angular velocity of the involved bodies is set equal

by conservation of momentum. Simulation results are shown in Section IV-D.

IV. PROTOTYPE

A. Mechanical Design

The linchpin of this design is the hip joint that pivotally connects each tread assembly to the chassis of the robot. With both motors mounted in the chassis, this joint independently transmits two coaxial torques: one to rotate the sprocket driving the treads and a second to rotate the tread assembly with respect to the chassis. Optical encoders (360 counts per revolution, CPR) are mounted within the chassis coaxially on both motor shafts. An additional high-resolution (5000 CPR) optical encoder is mounted on the outboard side of the tread assembly to measure the relative rotation of the tread shaft. The higher resolution encoder enables better low speed estimation (see Section IV-C for a detailed discussion).

The actuation of the two degrees of freedom of each hip joint enable the robot to perform its unique suite of maneuvers. A set of passive unactuated wheels is mounted on the end of the chassis opposite the main drive axles. The wheels prevent the chassis from dragging on the ground when the tread assemblies are rotated higher than the chassis.

The diameter of the tread sprocket was chosen to give over 25 mm of ground clearance for the chassis when in a horizontal configuration, in conjunction with the motor and gearbox choice to have sufficient torque to lift the weight of the robot, and a top speed in excess of 2.5 m/s (six body lengths per second). The traction between the treads and various ground surfaces is balanced between the need to grip while accelerating, and the need to slip while skid-steering. It is critical to maintain traction while balancing upright, where a tread slipping may cause the robot to fall. The off-the-shelf treads (manufactured by VEX Robotics) are made of acetal and have a spray rubber coating to increase traction. The treads are continuously supported underneath, any large normal force on the treads will be transferred to the structure of the tread assembly (also acetal). The power loss incurred by the smooth sliding contact between the treads and tread assemblies is included in the dynamic model (2). The position of the rear idler sprocket is adjustable to appropriately tension the treads.

Great pains were taken to minimize the part count, particularly the custom part count, and to reduce the number of machining operations per custom part. Off-the-shelf parts were used wherever possible to reduce manufacturing time. All but two of the custom parts are laser cut from sheets of acetal, with thread-tapping of some holes (thus avoiding the need for nuts) and press-fitting bearings being the only secondary machining operations. The remaining two custom parts are formed from stainless steel rod stock with simple operations on a lathe and milling machine. The symmetry of the design reduces the unique part count and many parts are orientation-agnostic, simplifying assembly. The 26 unique custom parts account for 73 pieces used in the assembly of each robot, with most parts being used in multiple places.

The design process included careful consideration of assembly time. The parts of the superstructure are quickly assembled with a series of interlocking tabs and slots (see Fig. 6), thereby

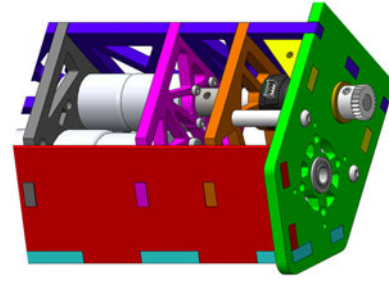


Fig. 6. View of the hip joint. Tab-and-slot construction simplifies assembly and reduces the number of screws required.

TABLE I
PARAMETERS OF THE PHYSICAL SYSTEM

Parameter	Value	Parameter	Value
m_C	3.662 kg	L_C	0.122 m
m_T	1.828 kg	L_T	0.126 m
m_S	0.090 kg	r	0.066 m
m	5.580 kg	ρ	0.011 m
μ_k	0.216	c_T	2.261 N-m
σ_S	4.127 N-m	ζ_S	0.187 N-m/s
σ_T	31.595 N-m	ζ_T	13.478 N-m/s
J_{SM}	0.002 kg-m ²	J_{TM}	0.645 kg-m ²
J_C	0.065 kg-m ²	J_T	0.022 kg-m ²
J_S	0.001 kg-m ²		

minimizing the number of mechanical fasteners needed and saving cost, weight, and assembly time. A team of five undergraduates working under our direction constructed 13 Switchblade robots from part manufacturing to final assembly in ten weeks. Many system parameters (masses, lengths, etc.) were directly measured or taken from the 3-D CAD model. Other parameters (stall torques, back EMF damping coefficients, rotational inertias, coefficients of friction) were determined empirically with an *in situ* characterization. We used a least squares algorithm similar to what is presented in [20]. Fundamental parameters of the physical system are listed in Table I.

B. Embedded Electronics

The Switchblade robot is built around the National Instruments sbRIO 9602 embedded controller. This board has both an FPGA and a PowerPC processor, which gives flexibility in handling both low-level high-speed tasks (such as reading optical encoders) and more complex control algorithms (as described in Section III) in real time, and is programmed using the LabVIEW graphical programming language, including the control design and simulation and robotics modules. Built-in ethernet coupled with a wireless ethernet adapter enables real-time wireless communication, debugging, and deployment of software. A 16-bit analog-to-digital converter reads the output of the analog accelerometer and gyroscope and monitors the battery voltage.

C. Sensing and Estimation

On the physical system, we are limited in what states can be observed directly with the onboard sensors. No external sensors (e.g., motion capture systems) are used. Thus, we design an

estimator to recreate the full state vector x . The robot is instrumented with only a two-axis MEMS accelerometer, a single-axis MEMS gyroscope, and four optical encoders. No camera, laser scanner, or additional nonproprioceptive sensor is required to traverse a step edge.

1) *Inertial Estimation*: The gyroscope can measure the rotational velocity of the chassis providing a direct measurement of $\dot{\theta}$. We apply a first-order digital low-pass filter with a cutoff frequency of 60 rad/s to remove high-frequency noise. The accelerometer outputs acceleration magnitudes in two orthogonal axes; by taking the arctangent of the two values, the gravity vector can be calculated, under the assumption that the body accelerations are small. Adding a first-order discrete low-pass filter attenuates noise as well as disturbances from body accelerations. This method yields an acceptable estimate of θ only at low frequencies.

Another method to estimate θ is to numerically integrate the output of the gyroscope. Integration error and thermal bias build over time, but can be eliminated using a first-order digital high-pass filter, thus creating an estimate of θ that is valid only at high frequencies. We create a complementary filter by choosing the accelerometer low-pass filter constant μ_{ALP} and integrated gyroscope high-pass filter constant μ_{GHP} such that they sum to unity, which allows the two measurements to be simply summed for a single estimate of θ valid across a wide frequency range [25]

$$\mu_{GHP} = \frac{1/\omega_c}{1/\omega_c + h}, \quad \mu_{ALP} = \frac{h}{1/\omega_c + h}.$$

We choose the crossover frequency $\omega_c = 0.5$ rad/s to tradeoff sensor errors, such as body accelerations and integration error.

2) *Encoder Velocity Estimation*: The optical encoders precisely measure the relative rotation between the tread sprockets and tread assemblies and between the chassis and the tread assemblies. Measuring the relative rotational velocity with the encoders is more difficult, particularly at low velocity. Counting the number of encoder ticks in a single time step h has unacceptable discretization noise when the speed is low because there are few to no encoder ticks per time step. A low pass or moving average filter may be applied to smooth the data, but it inherently adds delay, decreasing the bandwidth of the system which can cause the controller to fail.

An alternative method is to use a high-speed clock (e.g., an FPGA) to measure the time between subsequent encoder ticks. Using quadrature encoding and counting both rising and falling edges of both channels of the encoder can increase the resolution by a factor of four (as well as determine the direction of rotation), but this common method can actually introduce significant noise to the velocity measurement. In practice, the output of the A and B channels of the encoder are not ideal square waves, the high time is not equal to the low time for a constant speed. Further, the phase offset between the A and B channels is not exactly 90° , and varies between rising and falling edges. What is more consistent is the 360° period from like edge to like edge (e.g., rising A to rising A, falling B to falling B, etc.), assuming the encoder is mounted concentrically with the shaft.

By measuring the four different periods independently instead of only measuring the subperiods between subsequent edges of any type, we maintain the factor of four increase in resolution

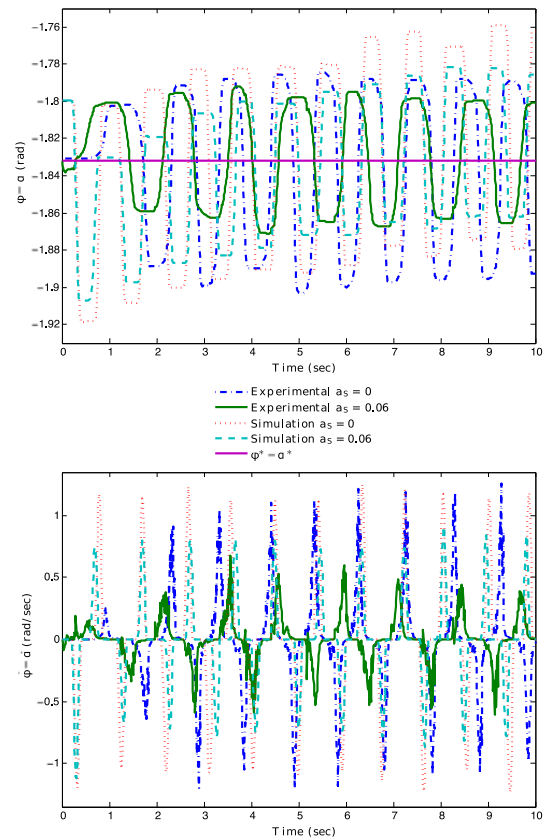


Fig. 7. Experimental and simulation results for stabilizing at $\alpha^* = \theta^* = \pi/2$ with and without friction compensation.

(and direction information) without introducing error from the asymmetry of the signals [26]. When a change of direction occurs, one channel of the encoder will output two opposite edges before the other channel outputs a single edge. In this case, it is impossible to measure the instantaneous speed because it is uncertain how far the encoder rotated before changing direction. However, we know that the average velocity between the two edges is zero because the net displacement over the time interval is zero, so we assume a measured speed of zero.

When the speed is great enough that at least two encoder edges pass in a single time step, it is possible to average the two (or more) time periods to smooth the velocity estimate. The smoothing effect is proportional to the speed. This is important to not only remove sensor noise from imperfections in the encoder disc, but also to smooth process noise. The disturbances caused by the sprocket teeth engaging and disengaging with the treads and other factors can induce significant noise if only measuring one encoder period.

By measuring the amount of time between encoder pulses, it is not possible to measure a speed of zero. Given the current time and the time since each of the last edges, it is possible to bind the actual speed by calculating the maximum speed that would not have yet produced an encoder edge. In practice, after the estimate decays sufficiently, it becomes negligible. Measuring the period of an encoder pulse is not an instantaneous measurement of speed, in fact it only yields the average speed over the interval of the last encoder pulse cycle. As speed decreases, this

TABLE II
RESULTS FROM HORIZONTAL STABILITY TESTS

	a_S	$\text{RMS}(\phi - \alpha) - (\phi^* - \alpha^*)$	$\text{RMS}(\dot{\phi} - \dot{\alpha})$	$\text{RMS } u_1$
Experimental	0	0.044	0.352	0.144
	0.06	0.028	0.169	0.167
Simulation	0	0.053	0.466	0.159
	0.06	0.038	0.326	0.161

measurement delays increases, decreasing the bandwidth of the system. Using a higher resolution encoder will improve performance, so long as the high-speed clock is still significantly faster ($\mathcal{O}(100)$) than the encoder pulse rate at the maximum shaft rotational velocity to avoid introducing discretization error. Encoder resolution is often a function of cost and size. Placing the encoder directly on the motor shaft, instead of the gearbox output shaft, greatly increases the resolution of the rotation of the output shaft (by a factor of γ , the gearbox ratio) without increasing the cost or size of the encoder, at the expense of not being able to measure the backlash at the gearbox output shaft.

3) *Sensor Fusion*: Given the estimates of $\{\theta, \dot{\theta}\}$ from the MEMS sensors and the encoder data, $\{\phi, \dot{\phi}, \alpha, \dot{\alpha}\}$ can be computed by simple addition. For the purposes of simulation, we mimic the output of the sensors by manipulating the state vector. For the encoders, we simply quantize the simulation state given the resolution of the encoder. For the MEMS gyroscope and accelerometer, we calculate what the sensor output would be given the current position, velocity, and acceleration add white noise (the variance of which is based on the manufacturer's specifications), and apply the complementary filter as described above.

D. Experimental Results

We first look at the performance of the system when stabilizing about a single equilibrium point, where $\alpha^* = \theta^* = \pi/2$. We use a narrow beam ($\rho = 0.011$ m) as a step edge surrogate for testing. We can compare the balancing behavior with and without the friction compensator (20) by choosing $a_S = 0$ or $a_S = 0.06$. In both cases, $b_S = b_T = 0.04$ and $a_T = 0$ (since $\alpha^* = \theta^*$ the friction between the chassis and tread assemblies does not need compensation). Due to backlash in the treads and uncertainty in the velocity estimate, the $\dot{\phi}$ gain is decreased 40% to eliminate chatter. As can clearly be seen in Fig. 7 and Table II, the controller with the friction compensator has significantly reduced variance from the reference command ($\phi^* - \alpha^*$) and the magnitude of the tread velocity ($\dot{\phi} - \dot{\alpha}$) peaks is greatly reduced without a significant increase in the control effort. The flat sections in the plot of $\phi - \alpha$ correspond to when the treads are in stiction. Please refer to the included video to see horizontal stability tests with ($a_S = 0.06$) and without ($a_S = 0$) the friction compensator. Discrepancies between simulation and experiment in Fig. 7 can be explained by differences in initial conditions and unmodeled effects of differences between the left and right drive trains.

Given the mass distribution of the robot, climbing the edge of a standard step, while kinematically possible (see Fig. 4), is not practically feasible. The distance between the center of

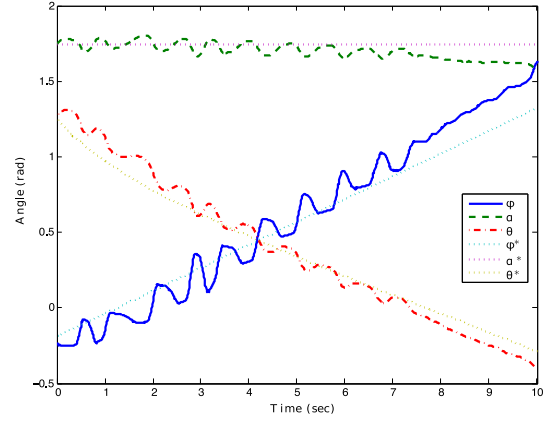


Fig. 8. Experimental results for traversing with $\alpha^* = 100^\circ$.

mass of the chassis and the tread assembly pivot point (L_C in Fig. 2) is less than half the length of the tread assemblies. This means that in order to shift the total center of mass from one end of the tread assemblies to the other, the inclination angle must be steep ($\alpha^* = 150^\circ$). By (15), a significant amount of the torque available from the tread motor is used by the feedforward term and not available for error regulation (in one direction). This problem is exacerbated as the height of the center of mass increases because more torque is required to recover from a given disturbance. The amount of stiction and friction between the treads and the tread assemblies further limits the control authority since the friction limits the amount of torque that can be used for correction, and there is zero control authority when in stiction. Worse, the normal force on the treads increases with α (2) which increases the friction and stiction. With the current prototype build, only a limited traverse is possible. Logged data from a 10 cm traverse at an inclination of 10° from horizontal ($\alpha^* = 100^\circ$) over 10 s is shown in Fig. 8, the maneuver is also shown in the included video. The oscillatory behavior is due to the treads entering and exiting stiction. The steady-state error is due to sensor and parameter errors.

V. CONCLUSION

We have presented a novel approach to the problem of stair climbing utilizing feedback control on a small mobile robot. Successful experimental results have been shown for a prototype traversing across a narrow beam. This method of stair climbing enables a relatively small robot (length scale on the same order as the height of a single step) to climb stairs without any external or expensive sensors and without any dedicated stair climbing hardware. As mentioned in the previous section, the mass distribution of the current robot is not optimized for climbing standard-sized stairs. It may be possible to improve performance with a different form of nonlinear controller, or an estimator that can detect when the treads are in stiction; however, greater performance gains could be achieved by decreasing the mass and shifting the center of mass of the chassis further from the tread assembly pivot point (alternatively, the tread assembly pivot point could move closer to the center of the tread assemblies). Stiction and friction could be significantly reduced by

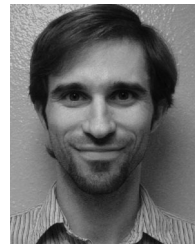
switching from discrete tread links to a timing belt and changing the design of the tread assemblies to avoid rubbing. All of these changes would act to increase the control authority of the system. In future work, the robot could perform automated parameter identification of a staircase (i.e., rise and run) from sensor data, such as from a camera or LIDAR. The robot could then calculate a trajectory and controller to climb the staircase using the algorithms outlined in this paper. One or more robots could be used to autonomously map a multistory building, such as in a fire fighting or urban warfare scenario. Modeling of the dynamics could also be expanded to three dimensions and the perching controller could be generalized to balance on uneven geometry, adaptively estimating the point of contact, possibly with additional sensors [27].

ACKNOWLEDGMENT

The authors would like to thank National Instruments for their generous support, Maxon motor for donating motors, and numerous members of the Coordinated Robotics Lab for their assistance in developing the Switchblade prototypes.

REFERENCES

- [1] DARPA Robotics Challenge. (2013). [Online]. Available: <http://www.theroboticschallenge.org>
- [2] SCHAFT Inc. (2013). [Online]. Available: <http://schaft-inc.jp>
- [3] Boston Dynamics. (2013). [Online]. Available: http://www.bostondynamics.com/robot_Atlas.html
- [4] B. Yamauchi, "PackBot: A versatile platform for military robotics," *Proc. SPIE*, vol. 5422, pp. 228–237, 2004.
- [5] C. X. Chen, M. M. Trivedi, and C. R. Bidlack, "Simulation and animation of sensor-driven robots," *IEEE Trans. Robot. Autom.*, vol. 10, no. 5, pp. 684–704, Oct. 1994.
- [6] E. Z. Moore, D. Campbell, F. Grimmering, and M. Buehler, "Reliable stair climbing in the simple hexapod RHex," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, pp. 2222–2227.
- [7] J. B. Morrell, J. M. Kerwin, D. L. Kamen, R. R. Ambrogi, R. J. Duggan, R. K. Heinzmann, and B. R. Key, "System and method for stair climbing in a cluster-wheel vehicle," U.S. Patent 6 311 794, Nov. 6, 2001.
- [8] S. D. Herbert, A. Drenner, and N. Papanikolopoulos, "Loper: A quadruped-hybrid stair climbing robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 799–804.
- [9] S. C. Chen, K. J. Huang, W. H. Chen, S. Y. Shen, C. H. Li, and P. C. Lin, "Quattroped: A legwheel transformable robot," *IEEE/ASME Trans. Mechatron.*, vol. 19, no. 2, pp. 730–742, Apr. 2014.
- [10] S. A. Stoeter and N. Papanikolopoulos, "Autonomous stair-climbing with miniature jumping robots," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 35, no. 2, pp. 313–325, Apr. 2005.
- [11] C. Schmidt-Wetekam, D. Zhang, R. Hughes, and T. Bewley, "Design, optimization, and control of a new class of reconfigurable hopping rovers," in *Proc. IEEE Conf. Decision Control*, 2007, pp. 5150–5155.
- [12] University of California San Diego Coordinated Robotics Lab. (2009). [Online]. Available: <http://robotics.ucsd.edu/iHop.html>
<http://www.youtube.com/watch?v=rUbMQzLjo2Y>
- [13] T. Takaki, T. Aoyama, and I. Ishii, "Development of inverted pendulum robot capable of climbing stairs using planetary wheel mechanism," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 5618–5624.
- [14] R. C. Luo, M. Hsiao, and T. W. Lin, "Erect wheel-legged stair climbing robot for indoor service applications," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 2731–2736.
- [15] J. Koh, D. Lee, S. Kim, and K. Cho, "Deformable soft wheel robot using hybrid actuation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 3869–3870.
- [16] N. Morozovsky, C. Schmidt-Wetekam, and T. Bewley, "Switchblade: An agile treaded rover," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 2741–2746.
- [17] C. Nie, X. P. Corcho, and M. Spenko, "Robots on the move: Versatility and complexity in mobile robot locomotion," *IEEE Robot. Autom. Mag.*, vol. 20, no. 4, pp. 72–82, Dec. 2013.
- [18] D. A. Wells, "A 'power function' for the determination of lagrangian generalized forces," *J. Appl. Phys.*, vol. 16, no. 9, pp. 535–538, Sep. 1945.
- [19] K. Iagnemma and C. C. Ward, "Classification-based wheel slip detection and detector fusion for mobile robots on outdoor terrain," *Auton. Robots*, vol. 26, no. 1, pp. 33–46, Jan. 2009.
- [20] N. Morozovsky, R. Moroto, and T. Bewley, "RAPID: An inexpensive open source dynamometer for robotics applications," *IEEE/ASME Trans. Mechatron.*, vol. 18, no. 6, pp. 1855–1860, Dec. 2013.
- [21] B. Bona and M. Indri, "Friction compensation in robotics: An overview," in *Proc. IEEE Conf. Decision Control Eur. Control Conf.*, 2005, pp. 4360–4367.
- [22] H. K. Khalil, "Exact feedback linearization," in *Nonlinear Systems*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1996, ch. 12, sec. 12.3.3, pp. 554–558.
- [23] A. E. Bryson, Jr., Y. C. Ho, *Applied Optimal Control*. New York, NY, USA: Taylor & Francis Group, 1975.
- [24] J. R. Dormand, and P. J. Prince, "A family of embedded Runge-Kutta formulae," *J. Comput. Appl. Math.*, vol. 6, no. 1, pp. 19–26, Mar. 1980.
- [25] W. T. Higgins, "A comparison of complementary and Kalman filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-11, no. 3, pp. 321–325, May 1975.
- [26] R. C. Kavanagh, "Improved digital tachometer with reduced sensitivity to sensor nonideality," *IEEE Trans. Ind. Electron.*, vol. 47, no. 4, pp. 890–897, Aug. 2000.
- [27] C. Schmidt-Wetekam, N. Morozovsky, and T. Bewley, "Inertial rotation center position estimation for a perching treaded vehicle," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 4097–4102.



the ASME.

Nicholas Morozovsky (M'11) received the B.S. degree from Carnegie Mellon University, Pittsburgh, PA, USA, in 2007, and the M.S. and Ph.D. degrees from the University of California, San Diego, CA, USA, in 2011 and 2014, respectively, all in mechanical engineering.

He has served as a robotics and 3-D printing consultant for multiple consumer-facing and business-facing companies. His research interests include mobile robot locomotion, mechanism design, embedded systems, and rapid prototyping. He is a member of



Thomas Bewley received the B.S./M.S. degree from Caltech, Pasadena, CA, USA, in 1989, the Diploma degree from von Karman Institute, Belgium, in 1990, and the Ph.D. degree from Stanford University, Stanford, CA, in 1998.

He directs the UCSD Flow Control and Coordinated Robotics Labs. He currently works at the intersection of semiautonomous agile robotics and the analysis, estimation, and forecasting of environmental flows using advanced control theory and numerical methods. His Coordinated Robotics Lab is developing an array of clever vehicles to achieve maximum agility with minimal complexity, and is coming to the mass market with a number of small toy and educational robotic vehicles in collaboration with WowWee Robotics. His Flow Control Lab is developing new algorithms for weather-forecasting class problems like state estimation and adaptive observation in environmental contaminant plumes, and *in situ* monitoring of developing hurricanes using buoyancy-controlled balloons.