Robotica: (2020) volume 38, pp. 159–170. © Cambridge University Press 2019
doi:10.1017/S0263574719000535

Autonomous Stairwell Ascent

B. Deniz Ilhan†*, Aaron M. Johnson‡ and D. E. Koditschek†

†*Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA.*
E-mail: kod@seas.upenn.edu
‡*Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA.*
E-mail: amj1@cmu.edu

(Accepted March 24, 2019. First published online: May 2, 2019)

SUMMARY
This paper documents autonomous multi-floor stairwell ascent by a legged robot. This is made possible through empirically deployed sequential composition of several reactive controllers, with perceptually triggered transitions. This composition relies on simplified assumptions regarding the robot's sensory capabilities, its level of mobility, and the environment it operates in. The discrepancies between these assumptions and the physical reality are capably handled by the intrinsic motor competence of the robot. This behavior is implemented on the legged RHex platform and experiments spanning 10 different stairwells with various challenges are conducted.

KEYWORDS: Autonomous robot; Stair climbing; Sequential composition; Self-manipulation.

1. Introduction

This paper reports on a task-level autonomy application where a legged robotic platform performs autonomous ascent over various multi-floor stairwells representing a diverse set of challenges (Fig. 1). Combined with the initial implementation presented in ref. [1], this work offers the first documented account of fully autonomous ascent over diverse stairwells composed of multiple flights of stairs with varying landing configurations. Similar to ref. [2], this is accomplished via simplified models regarding the environment the robot operates in, the sensors it is equipped with, and the level of autonomy it is capable of, abstracting away all the complexities handled by the robot's mechanical *preflexes*.[3,4] With the help of these assumptions, the task at hand is broken into a set of reactive controllers called by perceptually driven event triggers in a manner that empirically suggests (although is not yet formally shown to exhibit) a sequential composition of attractor basins.[5]

Experiments presented in this work span 10 stairwells in 4 different buildings, logging a legged robot autonomously ascending over 67 flights of well over 700 diversely proportioned stairs and landings. The platform used for these experiments is X-RHex,[6,7] a re-engineered version of RHex.[8] Its high power density, flexible sensor interface, and software Application Programming Interface (API) enabled this implementation of the commanded behavior in a very straightforward way. Taking advantage of its modular payload architecture, a light detection and ranging (LIDAR) sensor and an inertial measurement unit (IMU) are attached to the payload rails spanning the robot's top side and utilized for all the sensing needs for successful execution. These experimentation efforts are reported in Section 4, where the robot performance and reported failure modes are discussed in detail.

1.1. Motivation

The single flight stair climbing gait for RHex was first introduced in ref. [9], and as discussed in detail in ref. [1], this gait works reliably over a wide variety of typical human-scale staircases. Versatile

* Corresponding author. E-mail: bdeniz@seas.upenn.edu

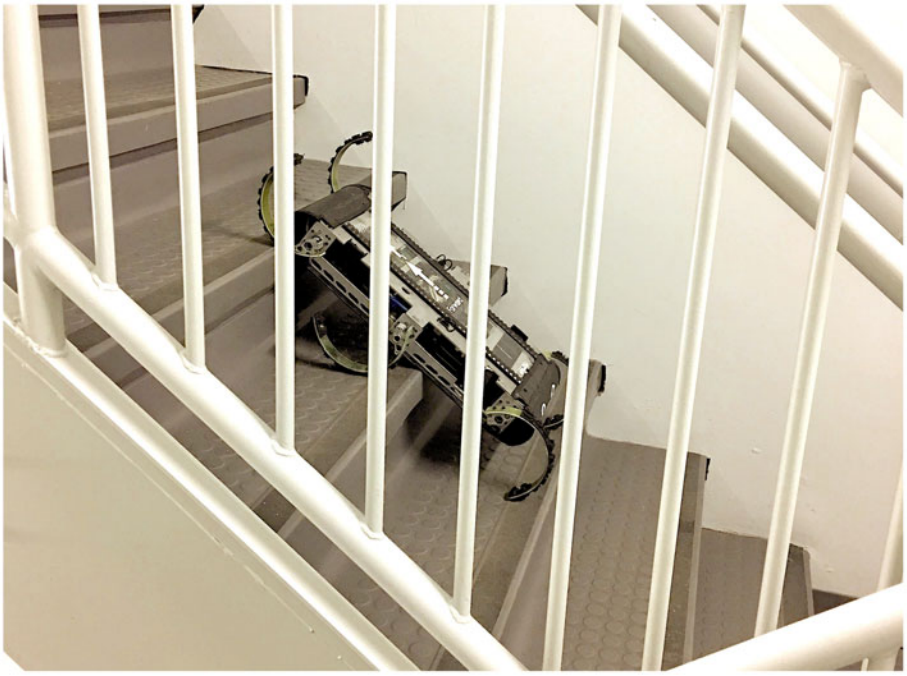

Fig. 1. The X-RHex robot climbing a stairwell.

robots capable of autonomous mobility in both indoor and outdoor settings have long been sought for remote operators executing urban search and rescue (USAR) and intelligence, surveillance, reconnaissance (ISR) operations.[10] Autonomous negotiation of multi-flight stairwells in indoor settings has specifically been long established as a very important yet challenging task for many existing man-portable mobile robots;[11] yet as discussed below in a comparative account of the more recent literature, surprisingly few advances targeting this capability have been reported. Such a capability can be especially useful when integrated into a multi-layer task navigation framework as discussed in ref. [2], enabling a *deliberative* task planner or a remote operator to focus only on high-level planning of a mission and delegate the task of traveling from one floor of a building to another to the lower level control authority presented in this work, facilitating its use in environments with little to no communication between the robot and the operator.[12]

1.2. Contributions and related work

Combined with ref. [1], the central contribution of this work is a complete account of autonomous ascent of general multi-floor stairwells. Its success over a variety of building interior styles is depicted in the data tables of Section 4. This is achieved via several modifications to ref. [1]. First of all, a more detailed descriptions of all the sensors utilized for implementing the resulting behavior are provided. Moreover, some of these sensor implementations are modified. Specifically, the pitch wiggle self-manipulation routine in Section 2.4.3 is updated to improve the quality and span of the pitch-up scan utilized for stair detection.

Similarly, drastic modifications are performed on the methodology for detecting stairs in Section 2.4.5, where the output of the pitch scan sensor is now treated as a depth image, and simple geometric processing is performed to detect a set of stairs. Furthermore, some refinements in the landing exploration behavior implementation in Section 3.2 are made. All these modifications resulted in improved performance: on a set of stairwells partially overlapping with their counterparts from ref. [1], only 12 behavioral problems over 731 stairs are reported, compared to 23 behavioral problems over 671 stairs.

Prior work on the problem of autonomous stair ascent can be grouped into three categories: ascent behavior over a single flight of stairs, algorithms for stair detection, and transitions between stairwells and landings.

https://doi.org/10.1017/S0263574719000535 Published online by Cambridge University Press

The first group of prior work focuses on the stair ascent behavior over a single flight of stairs.[13–23] Many are limited to a very few steps,[15,17–20,22,23] or a very specific stair or landing geometry.[14,16,17,22,23] In addition, most lack comprehensive experimentation to report on reliability.[13–17,19–23] This work on the other hand builds upon the stair ascent behavior from ref. [9] that reliably works on a variety of flights of stairwells and, combined with ref. [1], expands it to multiple flights of stairs.

The second group focuses on the detection of the stairs themselves.[13,14,16,21,24–30] The choice of sensor in this work—a LIDAR as opposed to a camera,[13,14,24] or a depth sensor[16,21,25–30]—differentiates how the sensory information about the stairs is generated and how this information is processed to accomplish the detection from these approaches.

The last group deals with autonomous transitions between flat surface walking and stair ascent under the control of an operator.[15,17,27,31] The transition approach presented in this work follows,[31] as discussed in Section 3.1.

The only reports found documenting empirical work on autonomy over multiple flights of stairs other than ref. [1] mention a few anecdotal successes[13] or assume a very specific, simple landing geometry.[14] This work intentionally targets a great diversity and reports in Section 4 on failure rates and causes in a detailed manner.

2. Robot and Task

This section presents simple models for the environment in which the robot is deployed, its degree of mobility therein, and the sensors with which it is equipped. The world model of the stair ascent task is complicated by the intermittent disappearance of the gradient beacon field (on flat landings) and the need to find specifically *marked* obstacles (flights of stairs) whereon a distinctly different gait yields robust ascent. The stair ascent behavior is accordingly complicated, and formal statements of correctness would have a stochastic character governed by the statistical properties of real stairwells. Although a formal demonstration of correctness lies beyond the scope of this paper, this section aims to present a precise enough account of all the modeling decisions to enable future analysis (when coupled with the description in the following section of the behavior that relies upon them).

2.1. World model

This section follows the steps of refs. [1] and [2], and introduces a very simple model of the *terrain* the robot is operating on and thus abstracts away many of the issues that may arise from operating on a complex engineered environment such as stairwells. This model relies on two assumptions. First, there are no obstacles present over the stairs[1], and the robot's stair climbing gait[9] can reliably traverse various stair designs. In addition, the landings are composed of simple polygonal floor plans with walls and, most of the time, the only openings are sets of stairs, up to two of them, one connecting to the lower floor and/or another connecting to the upper floor. This simplification of the world model is appropriate for a platform like RHex since small obstacles such as debris or uneven surfaces do not pose any problems for the robot's standard walking gait. In addition, the violations to the polygonal landing assumption will not necessarily pose any threat to robot operation as there is no reason for nonzero curvatures to pose any problems for the behavior introduced in Section 3.

A *stairwell*, $\mathcal{S}$, is defined to be a piecewise constant terrain where the *terrain*, defined in ref. [2], is represented by some unknown height function, $h : \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$. A constant component, $\mathcal{U}_i$, over this stairwell is called a *landing*. A landing is surrounded by obstacles such as walls and cliffs, and potentially another component, $\mathcal{V}_i$, called *stairs* (as described in Section 2.4.5) that connect it to the next (higher level) landing. Note that, in this definition, a *cliff* can be either the previous set of stairs, $\mathcal{V}_{i-1}$, connecting the landing to a lower level, or missing walls, banisters, so on, that represent hazardous conditions for safety of the platform.

2.2. Robot model

This work utilizes two different models depending which stairwell component the robot is currently operating.

[1]Note that the robot is capable of steering over a flight of stairs. An obstacle avoidance scheme similar to ref. [2] could be employed for obstacle-strewn stairwells.

https://doi.org/10.1017/S0263574719000535 Published online by Cambridge University Press

For the operations over a landing, the robot's standard gait (alternating tripod[8]) is assumed to reduce to the target dynamics (or *template*[32]) of a horizontal plane kinematic unicycle,[2]

$$\dot{\mathbf{q}} = B(\theta)\mathbf{u}_{ku}; \quad B(\theta) := \begin{bmatrix} \mathbf{n}(\theta)\mathbf{e}_1^T \\ \mathbf{e}_2^T \end{bmatrix}, \tag{1}$$

with $\mathrm{I} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 \end{bmatrix} := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. The robot state, $\mathbf{q} := (\mathbf{p},\ \theta) \in SE(2)$ contains its position on the plane, $\mathbf{p} \in \mathbb{R}^2$ and its heading, $\theta \in \mathbb{S}^1$. $\mathbf{n}(\theta)$ denotes the unit vector representing the robot's heading, whereas the control input $\mathbf{u}_{ku} \in \mathbb{R}^2$ is composed of translational and rotational velocity components, respectively.

Over the stairs, the stair climbing gait[9] is represented as a scalar point particle tracking the single-dimensional gradient defined by the slope of the stairs.

2.3. Task model

The task of *autonomous stairwell ascent* requires that the robot locomotes from any initial position and orientation over a stairwell to some landing with no *stair* boundaries.

2.4. Sensor models

This section provides a list of abstract sensor models used for implementing the autonomous stairwell ascent behavior. These sensors are a succession of exteroceptive sensors that can be realized through the use of an IMU and a LIDAR hardware unit mounted on a legged robot.

2.4.1. Depth sensor. The depth sensor is an abstract map,

$$\sigma_E : SE(2) \times B \times P \to R \tag{2}$$

that returns from each position and heading in the plane, $(\mathbf{p}, \theta) \in SE(2)$, bearing angle, $\beta \in B := [-\beta_u, \beta_u]$, body pitch, $\phi \in P := [\phi_l, \phi_u]$, and a distance, $\rho \in R := [0, \rho_u]$.

For the implementation, the output from a fixed LIDAR[2] unit is utilized to realize this depth map. The arc extends roughly $\pm 120^\circ$ off center. The distance profile corresponds to the first depth at which the LIDAR unit records a return. The LIDAR unit cannot detect beyond a distance of $\rho_u := 4m$, to which the *infinite* reading of its maximum depth scale is calibrated.

2.4.2. Gap sensor. The gap sensor is an abstract map,

$$\sigma_G : SE(2) \to B \tag{3}$$

that returns for each position and orientation at which the robot is pointing, the center, $\sigma_G(x, y, \theta) = \xi$ of an arc segment $[\xi - S, \xi + S] \subset B$, and a *window* within which the interval depth is maximum

$$\xi := \underset{\beta_l + S \le \tau \le \beta_u - S}{\operatorname{argmax}} I[\tau, S], \tag{4}$$

with,

$$I[\tau, S] := \min_{\tau - S \le \beta \le \tau + S} \frac{\sigma_E(x, y, \theta, 0, \beta)}{(1 - K)\cos^6(\beta - \tau) + K}, \tag{5}$$

where the power term over the cosine function introduces the bias toward lower bearing differences to emulate the search for a rectangular opening on the robot's path.

2.4.3. Pitch scan sensor. The pitch scan sensor, $\sigma_P : SE(2) \times B \times P \to R \times B \times P$, is defined as

$$\sigma_P(\mathbf{p}, \theta, \phi_l, \phi_u) := \left\{ (\sigma_E(x, y, \theta, \phi, \beta),\ \beta,\ \phi) \ : \beta \in B, \phi \in [\phi_l,\ \phi_u] \subset P \right\} \tag{6}$$

and is implemented by running the depth sensor at each bearing angle within the field of view and pitch angle achieved via a coordinated motion of the legs—a *pitch wiggle* self-manipulation.[1,33,34]

[2]Hokuyo URG-04LX-F01, http://www.hokuyo-aut.jp/

https://doi.org/10.1017/S0263574719000535 Published online by Cambridge University Press

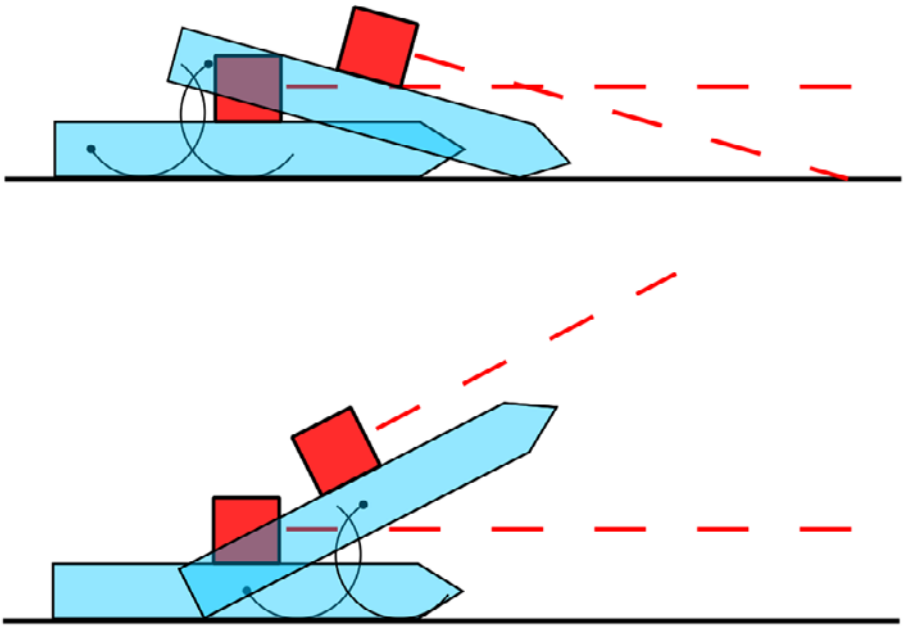

Fig. 2. The pitch wiggle behavior for up and down scans, with inactive legs removed for clarity.

The *pitch wiggle* is a sensorimotor routine utilizing the planar LIDAR to generate a depth image. For a horizontally placed LIDAR unit, even for several scans combined, there is no perceived differences between a stairwell and a wall. To generate a depth image, instead of attaching the LIDAR unit to a motor to tilt it, RHex morphology enables manipulation of its body pitch to a range of angles, where an IMU[3] unit is utilized to measure these angles. As discussed in refs. [1, 34], this routine, as depicted in Fig. 2, produces a large variation in body pitch (either up or down) with no internal forces or toe slip.

2.4.4. Cliff sensor. The cliff sensor, $\sigma_C : SE(2) \times B \times P \to \{0,\ 1\}$, is the composition $\sigma_{CD} \circ \sigma_P$. The pitch scan sensor, σ_P, is pitched through a downward interval ($\phi_l < \phi_u < 0$) to scan a mid distance rectangular region on robot's path. The cliff detection sensor

$$\sigma_{CD} : R \times B \times P \to \{0,\ 1\} \tag{7}$$

compares the results from σ_P with predicted range values from current pitch and bearing angles and returns a binary value based on the persistence of segments with extreme negative error. It contains two stages. In the first stage, ground range prediction error

$$\sigma_{GE}(\rho,\ \beta,\ \phi) := \mu(\beta,\ \phi) - \rho \tag{8}$$

is computed for every $(\rho,\ \beta,\ \phi) \in \sigma_P$ through the ground range prediction function $\mu : B \times P \to R$ as

$$\mu(\beta,\ \phi) := \frac{0.5\, l \tan(-\phi) + h_s}{\tan(-\phi)} \cdot \frac{1}{\cos\beta}, \tag{9}$$

where, assuming that LIDAR is located at the geometric center, l is the length of robot's body and h_s is the total height of the LIDAR and robot body. After a unidirectional threshold, a binary value based on the persistence of segments with extreme negative error is returned.

2.4.5. Stair sensor. The stair sensor, $\sigma_S : SE(2) \times B \times P \to R \times B \times \mathbb{S}^1 \times \{0,\ 1\}$, is the composition $\sigma_{SD} \circ \sigma_P$. The pitch scan sensor, σ_P, is pitched through an upward interval ($0 < \phi_l < \phi_u$).

The stair detection sensor $\sigma_{SD} : R \times B \times P \to R \times B \times \mathbb{S}^1 \times \{0,\ 1\}$ returns the range ρ_S, bearing β_S and normal angle θ_S of the stairwell and a binary variable c_S indicating if the sensor is confident

[3]Microstrain 3DM-GX2, http://www.microstrain.com/

https://doi.org/10.1017/S0263574719000535 Published online by Cambridge University Press

about this detection. It outputs zero if it cannot detect stairs. It is implemented in three stages. To detect and extract output parameters, a stairwell is modeled as a set of vertical plane segments with increasing horizontal offset where offset difference between successive plane segments is within a predefined interval $[d_l, d_u] \subset R$. At the first stage, a line segment extractor $\sigma_{LS}(\sigma_P) := \{p^i_{\mathcal{L}}\}$ finds and parameterizes line segments,

$$\mathcal{L}^i := \{(\rho, \beta, \phi) \in \sigma_P : \rho \cos(\beta) = \rho \sin(\beta) a^i + b^i, \phi = \phi^i\}, \tag{10}$$

on the LIDAR scanning plane for every pitch angle. A line segment is represented with five parameters: pitch angle ϕ^i, bearing interval boundaries β^i_l, β^i_u, normal angle $n^i = atan(-a^i)$, and horizontal offset $d^i = b^i \cos \phi^i$, where

$$p^i_{\mathcal{L}} := \left(\phi^i, \beta^i_l, \beta^i_u, n^i, d^i\right). \tag{11}$$

Once all the line segments are extracted and parameterized, vertical plane segment extractor $\sigma_{PS}(\sigma_{LS}) := \{p^j_{\mathcal{P}}\}$ groups these line segments into vertical plane segments

$$\mathcal{P}^j := \left\{p^k_{\mathcal{L}} \in \sigma_{LS} : \left[\beta^k_l, \beta^k_u\right] \cap \left[\beta^{k+1}_l, \beta^{k+1}_u\right] \neq \emptyset, n^k = n^j, d^k = d^j\right\} \tag{12}$$

by comparing individual bearing angle intervals, normal angles, and horizontal offsets, and performs a parameterization. A plane segment is represented by six parameters: pitch interval boundaries ϕ^j_l,ϕ^j_u, total bearing interval boundaries $\beta^j_l = \min_k \beta^k_l$, $\beta^j_l = \max_k \beta^k_u$, normal angle n^j, and horizontal offset d^j, where

$$p^j_{\mathcal{P}} := \left(\phi^j_l, \phi^j_u, \beta^j_l, \beta^j_u, n^j, d^j\right). \tag{13}$$

Finally, the stair extractor $\sigma_{SE}(\sigma_{PS}) := p_{\mathcal{S}}$ returns the range, bearing and heading angles of the stairwell and a binary confidence variable if detected. It outputs zero otherwise. It first extracts a stair candidate

$$\begin{aligned}\mathcal{S} := \big\{ & p^k_{\mathcal{P}} \in \sigma_{PS} : n^k = n_S,\ d^{k=0} = \rho_S, d_l \leq d^{k+1} - d^k \leq d_u, \\ & [\phi^k_l, \phi^k_u] \cap [\phi^{k+1}_l, \phi^{k+1}_u] = \emptyset, [\beta^k_l, \beta^k_u] \cap [\beta^{k+1}_l, \beta^{k+1}_u] \neq \emptyset\big\}\end{aligned} \tag{14}$$

by comparing pitch and bearing intervals, normal angles, and horizontal offsets. A stairwell is represented by four parameters: stair distance ρ_S, stair central bearing angle β_S, stair heading $\theta_S = n_S + \theta$, and a binary confidence indicator c_S that is nonzero if minimum pitch angle $\phi^{k=0}_l$ and absolute bearing angle $|\beta_S|$ are both within some confidence intervals

$$p_{\mathcal{S}} := (\rho_s, \beta_S, \theta_S, c_S). \tag{15}$$

Figure. 3, composed of six images, depicts the actual implementation, where the first stage of the filter contains two preprocessing steps. Upper left represents the raw scan acquired from the depth sensor via an upward pitch scan, whereas upper right shows the result of a filtering process, where beginning from the lowest pitch angle, any infinite reading for a specific bearing is replaced by the reading for the same bearing from the lower pitch angle scan. Middle left is the output of a 1D edge detector employed to segment individual pitch angle scans into intervals. The horizontal line fitting as the first stage and vertical grouping of line segments for vertical plane fitting are all applied on this output. Middle right shows the output of the second stage, where individual vertical plane segments extracted through this process are represented with their average range value. Lower left and lower right both represent the output of the last stage, where a stairwell candidate is detected.

3. Autonomous Stairwell Ascent

Unlike the task of autonomous hill ascent[2] which is implemented as a single control law, the task of autonomous multi-flight stairwell ascent requires a hybrid system composed of several reactive controllers required by increased complexity in the sensorimotor loops the robot uses to detect stairwells,

https://doi.org/10.1017/S0263574719000535 Published online by Cambridge University Press

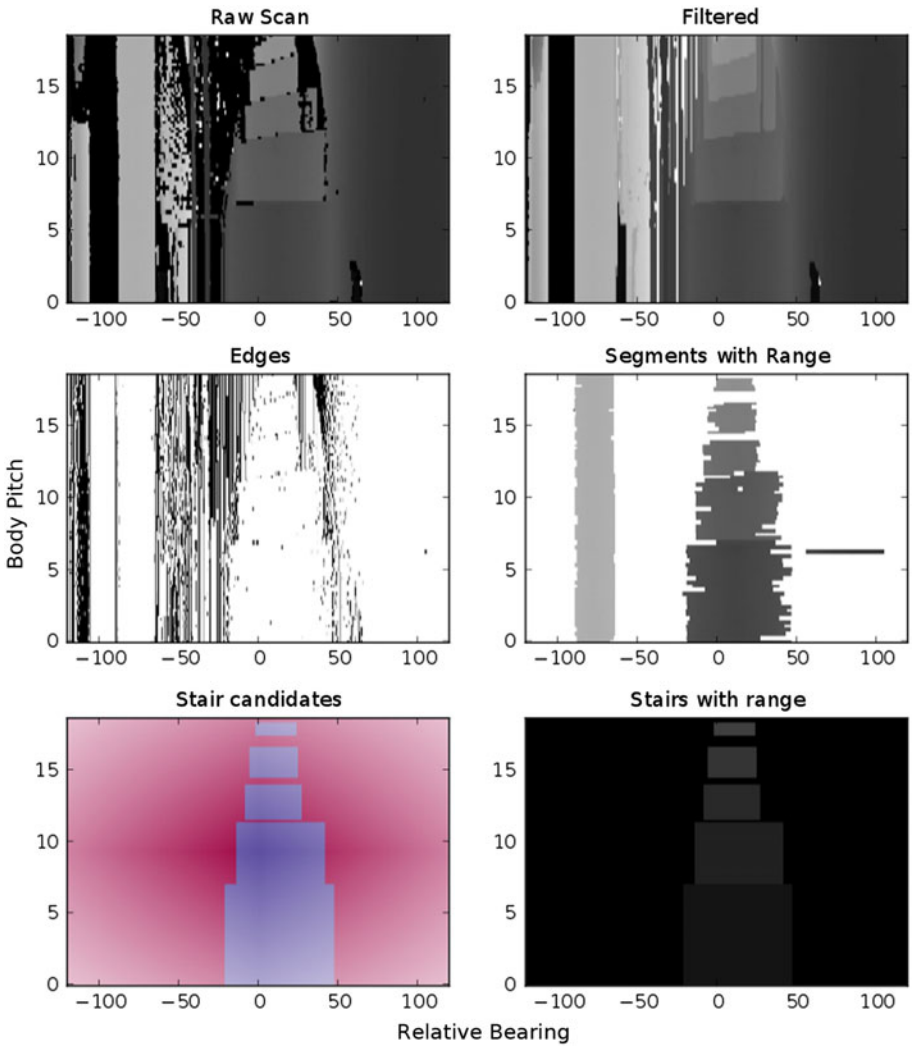

Fig. 3. Implementation details of the stair sensor. For all the graphs, the vertical axis denotes the body pitch and the horizontal axis denotes relative bearing angle in degrees. The top two graphs contain the raw readings and the output of a simple filter.

boundaries, and other hazards. Not all of these closed loop dynamics admit well-defined attractors and basins necessary for a formal implementation of sequential composition.[5] Instead, the task is implemented through systematic pre-image backchaining[35] and an implementation following the formal framework of ref. [5] is left for future work.

The overall task is implemented via two main behaviors, stair climbing and landing exploration. As summarized in Fig. 4, the robot executes a series of controllers backchained in a roughly cyclic pattern until it reaches the top of the final flight of stairs. The two behaviors and accompanying control routines are presented as follows.

3.1. The stair climbing behavior

As explained in detail by ref. [1], once RHex is fully engaged on a single flight of a stairwell, the overall open loop climbing gait reliably works as if the platform is guided by a steepest ascent controller, as devised in ref. [2]. Though, for a successful climb, when the robot detects a set of stairs, it first needs to enter the domain of the stair climbing controller. This implementation utilizes a transition routine previously presented in ref. [31] with some modifications from ref. [36]. Ref. [31] shows that this routine is empirically reliable. The robot detects the end of a flight of stairs via robot body pitch and executes the final transition to landing exploration, consisting of a couple steps forward.

https://doi.org/10.1017/S0263574719000535 Published online by Cambridge University Press

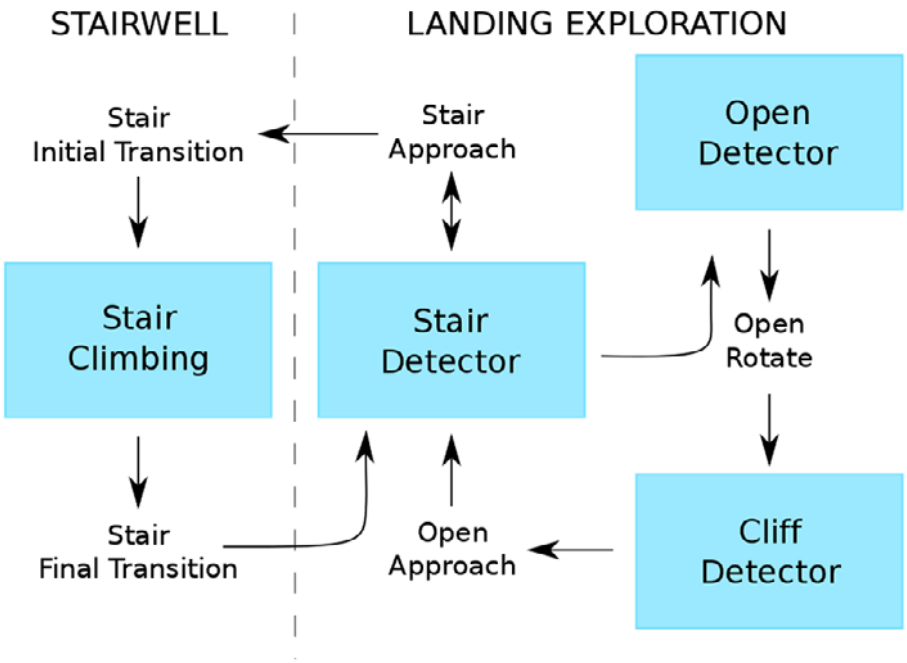

Fig. 4. Flow chart describing autonomous stair climbing.

3.2. Landing exploration behavior

Once the robot climbs through a flight of stairs and reaches a new landing, a sequence of controllers (as summarized in Fig. 4) is activated to drive it out of the prior goal set (i.e., the sensed zero-grade event that triggered the stair exit controller) and into the basin of the next as follows.

3.2.1. Stair detector.

- **Stair detector**
  This controller first calls the stair sensor, σ_S, and returns $(\rho_S, \beta_S, n_S, c_S)$ (Section 2.4.5). For nonzero output, this controller performs an open loop move to the relative pose $(\rho_S, \beta_S, \theta_S - \theta)$. If $c_S = 1$, the robot transitions into stair climbing behavior. Otherwise, it transitions back to σ_S for further investigation.
  If σ_S returns 0, the robot switches to the open detector controller.
- **Open detector**
  By calling the gap sensor σ_G (Section 2.4.2), this controller picks the *most open* bearing angle. At the beginning of each landing, the sign of this bearing angle is declared as the preferred direction to be used in case of future conflicts.
  If no suitably open bearing angle is available (if $\sigma_E \circ \sigma_G(x, y, \theta) < 1m$, i.e., the robot is in a corner), the robot simply rotates by 90° through the preferred direction and transitions back to the stair detector. In the presence of a suitably open bearing, the robot rotates to this angle and switches to the cliff detector controller.
- **Cliff detector**
  This controller first runs the cliff detector sensor σ_C (Section 2.4.4) to ensure it will not fall by pursuing this new heading. If this controller returns 0, robot walks for up to 1 m; otherwise it rotates back through the preferred direction and transitions back to the stair detector controller.

The behavior presented in section is not designed to be particularly efficient. On the contrary, this behavior relies on simple sensorimotor routines combining limited sensing capabilities with the robot's mechanical competence. Thus, it seems unreasonable to expect any deterministic guarantees that the robot can reach the basin of the next stairwell ascent controller (i.e., the first steps of the next upward stairs) through this sequence of controllers. Empirically, though, the data show that this behavior finds the subsequent stairwell with very high probability as landings are generally metrically small and topologically simple.

https://doi.org/10.1017/S0263574719000535 Published online by Cambridge University Press

4. Experimental results

The autonomous multi-flight stairwell ascent implementation is tested over 10 different stairwells in 4 buildings of the School of Engineering and Applied Science in the University of Pennsylvania. These results are summarized in Table I. This summary lists some characteristics of these stairwells, such as average rise and run lengths for the stairs, number of stairs, number of flights, landing styles, and sizes. It reports the total time it took for the robot to finish the stairwells, number of scans it performed, and any faults recorded. Following ref. [1], this summary makes a distinction between *behavior faults* (caused by shortcomings in the algorithm or the sensorimotor capabilities) and *robot faults* (mechanical, electrical, and communication failures). Out of the 10 stairwells the experiments were conducted on, only 2 met the world model requirements, with all solid walls and no openings other than stairwells. Even though the rest of these stairwells violated the assumptions in different ways, the robot successfully climbed through them. The implication of this performance is that the simplified model for the stairwells presented in this work successfully captures some general characteristics for a surprisingly wide variety of structures.

The control routines, sensory capabilities, and triggers forming the task-level autonomy developed for these experiments have been all implemented in Python[4] on a remote operator computer. While all these behaviors could be implemented directly on the robot, the use of this network abstraction layer has greatly sped up behavior development. However, occasional network glitches introduced experimental errors, as documented below, and could be mitigated in future versions of these behaviors.

Overall there were 12 behavior faults. The robot had only two false positives on stair detection throughout 67 flights of stairs. In particular, one of these two failures occurred because the specific landing had a window whose frame combined with the wall fit the stairwell model described in Section 2.4.5. The other failure could be avoided by cropping out small pitch angles as they managed to create enough features to mislead the plane segment extractor. Similarly, there were only two wall collision-based failures and both happened on stairwell number 5 where the laser scanner could see through the mesh walls and detect open space even though the mesh is actually an obstacle to the robot, leading to collisions that in turn precipitated faults requiring operator intervention. Cliff detection thresholds were rather conservative during the experiments to avoid any false positives which resulted in two possible cliff falls avoided by operator intervention. The remaining six behavior failures occurred during initial stair transitions. These could be avoided by more extensive sensor integration which is out of the scope of this work.

In addition, there were 21 robot faults. The majority of these arose from a leg failing to respond (8 times) and from network communication issues (9 times). The former issue was caused by known power distribution issues. It was partially addressed in the midst of experimentation and it is expected to be fully resolved in very near future. Additionally, there were three LIDAR failures each of which happened due to overheating. These failures resulted in low-quality readings which were addressable via power cycling the LIDAR. In the future, these failures can be fully avoided through simple heat dissipation solutions. Lastly, there was a single IMU failure due to a loose USB cable preventing the robot from detecting the end of the stairs.

To summarize, the autonomous multi-floor ascent behavior resulted in the robot climbing a total of 731 stairs in 67 flights. Over 5 h of testing, it encountered only 12 behavioral faults. Moreover, the mechanical competency of the legged platform overcame several incidents on almost all stairwells, such as legs hitting a wall or the robot engaging the stairwell at a bad angle. These incidents could otherwise be considered faults as they would require human intervention.

5. Conclusion

This paper presents an autonomous behavior, autonomous multi-flight stairwell ascent, composed of several sensorimotor and control loops stitched together via perceptually triggered transitions, loosely following the formal notion of sequential composition.[5] This behavior is not designed to be efficient or optimal but rather to exploit limited sensory availability in combination with the mechanical competence of the underlying platform. Extensive experimentation demonstrates its empirical robustness, which relies on the underlying mechanical competency of the physical platform.

[4]Python Programming Language, http://www.python.org/

https://doi.org/10.1017/S0263574719000535 Published online by Cambridge University Press

Table I. Ten indoor stairwell climbing behavior trials covering 731 stairs in 67 flights with a total of 12 behavioral problems. World model violations are briefly described. Rise, Run, and Landing size dimensions are given in centimeters (cm). Scans column contains two numbers; Stair Scans and Cliff Scans. Behavior faults are categorized as (S)tair Detection, (C)liff Detection, Stair (T)ransition, and (W)all Collision. Robot faults fall into four categories; (N)etwork Communication, (L)eg Failures, (L)I(D)AR Failures, and (I)MU Failures.

#	Violation	Rise (cm)	Run (cm)	Landing	Landing size (cm × cm)	# Flights	# Stairs	Time (h:min:s)	# Scans (stair, cliff)	Behavior	Robot
1	-	15.3	28.0	Straight	189 × 150	2	11	0:01:51	2, 0	-	-
2	-	15.3	28.0	Straight	327 × 150	2	11	0:03:01	3, 1	-	-
3	Glass	17.4	29.6	Straight	192 × 143	2	27	0:02:27	2, 0	-	-
4	Glass	16.7	26.9	Mixed	256 × 277	3	25	0:07:20	7, 4	-	-
5	Various	17.5	31.4	U-left	768 × 653	6	81	0:50:05	47, 36	1S	1N, 2L
6	Window	18.2	26.3	U-left	486 × 222	7	60	0:25:25	33, 22	1S, 1T	2N, 2L
7	Glass	16.2	28.5	U-left	471 × 252	10	111	1:03:25	51, 36	1C	3N, 3L
8	Glass	17.3	27.2	U-left	349 × 156	10	112	0:54:40	55, 39	2T, 1C	1N, 1LD, 1L
9	Mesh	17.3	27.2	Mixed	293 × 137	11	112	0:44:54	44, 26	1T, 2W	2N, 1LD
10	Heater	17.5	26.0	U-left	228 × 122	14	181	1:00:59	49, 27	2T	1LD, 1I

Similar to ref. [2], this behavior can become a part of a multi-layer navigation system, where the high-level authority can start this as a reactive layer behavior that can take the robot autonomously from floor A to floor B, without the need of a global planner carefully devising policies every step of the way. To better facilitate this vision, there are several modest improvements that could be pursued as the next step. The stair climbing behavior can be endowed with descent capability (as in ref. [37] via ref. [38]), as well more deliberative obstacle avoidance (as in ref. [39]). Lastly, this approach to task encoding and execution can be combined with the perceptual capabilities developed in ref. [27] for faster landing exploration.

Acknowledgments

This work has been supported in part by a DARPA MaxMobility Seedling grant and the Army Research Laboratory under Cooperative Agreement Number W911NF-10-2-0016, and in part by the National Science Foundation under NSF CDI-II Number 1028237. The authors would like to thank Avik De, Cameron Cogan, Jorge Raul Martinez, Joseph Cato, Justin Starr, Mike Choi, and Pouria Sanjari for providing assistance during the experiments.

References

1. A. M. Johnson, M. T. Hale, G. C. Haynes and D. E. Koditschek, "Autonomous legged hill and stairwell ascent," **In**: *IEEE International Workshop on Safety, Security, & Rescue Robotics, SSRR* (2011) pp. 134–142.
2. B. D. Ilhan, A. M. Johnson and D. E. Koditschek, "Autonomous legged hill ascent," *J. Field Robot.* **35**(5), 802–832 (2018).
3. I. E. Brown and G. E. Loeb, *A Reductionist Approach to Creating and Using Neuromusculoskeletal Models* (Springer, New York, USA, 2000) p. 148.
4. P. Holmes, R. J. Full, D. E. Koditschek and J. Guckenheimer, "The dynamics of legged locomotion: Models, analyses, and challenges," *SIAM Rev.* **48**(2), 207–304 (2006).
5. R. R. Burridge, A. A. Rizzi and D. E. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *Int. J. Robot. Res.* **18**(6), 534–555 (1999).
6. K. C. Galloway, G. C. Haynes, B. D. Ilhan, A. M. Johnson, R. Knopf, G. Lynch, B. Plotnick, M. White and D. E. Koditschek, "X-RHex: A Highly Mobile Hexapedal Robot for senzsorimotor Tasks," Technical Report (University of Pennsylvania, 2010).
7. G. C. Haynes, J. Pusey, R. Knopf, A. M. Johnson and D. E. Koditschek, "Laboratory on legs: an architecture for adjustable morphology with legged robots," **In:** *Unmanned Systems Technology XIV*, vol. 8387 (SPIE, 2012) p. 83870W.
8. U. Saranli, M. Buehler and D. E. Koditschek, "RHex: A simple and highly mobile hexapod robot," *Int. J. Robot. Res.* **20**(7), 616–631 (2001).
9. E. Moore, D. Campbell, F. Grimminger and M. Buehler, "Reliable Stair Climbing in the Simple Hexapod 'RHex'" **In**: *IEEE International Conference on Robotics and Automation, 2002. Proceedings, ICRA'02.* vol. 3 (IEEE, 2002) pp. 2222–2227.
10. A. Birk and S. Carpin, "Rescue robotics – a crucial milestone on the road to autonomous systems," *Adv. Robot.* **20**(5), 595–605 (2006).
11. R. Ray, B. Bepari and S. Bhaumik, "On design and development of an intelligent mobile robotic vehicle for stair-case navigation," **In**: *Intelligent Autonomous Systems* (D. Pratihar and L. Jain, eds.), vol. 275 (Springer, Berlin/Heidelberg, 2010) pp. 87–122.
12. R. R. Murphy, S. Tadokoro, D. Nardi, A. Jacoff, P. Fiorini, H. Choset and A. M. Erkmen, "Search and rescue robotics," **In**: *Springer Handbook of Robotics* (B. Siciliano and O. Khatib, eds.) (Springer, Berlin, Heidelberg, 2008) pp. 1151–1173.
13. Y. Xiong and L. Matthies, "Vision-Guided Autonomous Stair Climbing," **In**: *IEEE International Conference on Robotics and Automation, ICRA'00*, vol. 2 (IEEE, 2000) pp. 1842–1847.
14. S. Steplight, G. Engal, S.-H. Jung, D. B. Walker, C. J. Taylor and J. P. Ostrowski, "A Mode-Based Sensor Fusion Approach to Robotic Stair-Climbing," **In**: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2000) pp. 1113–1118.
15. P. Ben-Tzvi, S. Ito and A. A. Goldenberg, "A mobile robot with autonomous climbing and descending of stairs," *Robotica* **27**(2), 171–188 (2009).
16. R. C. Luo, M. Hsiao and T.-W. Lin, "Erect Wheel-Legged Stair Climbing Robot for Indoor Service Applications," **In**: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* IEEE (2013) pp. 2731–2736.
17. R. Morales, J. Somolinos and J. Cerrada, "Dynamic control of a reconfigurable stair-climbing mobility system," *Robotica* **31**(2), 295–310 (2013).
18. S. H. Turlapati, M. Shah, S. P. Teja, A. Siravuru and S. V. Shah, "Stair Climbing using a Compliant Modular Robot," **In**: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE (2015) pp. 3332–3339.

19. N. Morozovsky and T. Bewley, "Stair climbing via successive perching," *IEEE/ASME Trans. Mechatron.* **20**(6), 2973–2982 (2015).
20. 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)*, IEEE (2017) pp. 1080–1087.
21. J. Guo, J. Shi, W. Zhu and J. Wang, "Approach to Autonomous Stair Climbing for Tracked Robot," **In**: *2017 IEEE International Conference on Unmanned Systems (ICUS)*, IEEE (2017) pp. 182–186.
22. S. Caron, A. Kheddar and O. Tempier, "Stair climbing stabilization of the HRP-4 humanoid robot using whole-body admittance control," **In**: *IEEE International Conference on Robotics and Automation*, IEEE (2019)
23. K. Sasaki and K. Suzuki, "Active rotary-legs mechanism for stair-climbing mobility vehicle," *IEEE Robot. Automat. Lett.* **3**, 2237–2244 (2018).
24. S. Se and M. Brady, "Vision-Based Detection of Staircases," **In**: *Fourth Asian Conference on Computer Vision ACCV*, vol. 1 (2000) pp. 535–540.
25. J. A. Delmerico, D. Baran, P. David, J. Ryde and J. J. Corso, "Ascending Stairway Modeling from Dense Depth Imagery for traversability Analysis," **In**: *2013 IEEE International Conference on Robotics and Automation* (May 2013) pp. 2283–2290.
26. A. Sinha, P. Papadakis and M. R. Elara, "A Staircase Detection Method for 3D Point Clouds," **In**: *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV),* IEEE (2014) pp. 652–656.
27. 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, 2015) pp. 94 680B–8.
28. T. Westfechtel, K. Ohno, B. Mertsching, D. Nickchen, S. Kojima and S. Tadokoro, "3d Graph Based Stairway Detection and Localization for Mobile Robots," **In**: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* IEEE (2016) pp. 473–479.
29. A. Perez-Yus, D. Gutiérrez-Gómez, G. Lopez-Nicolas and J. Guerrero, "Stairs detection with odometry-aided traversal from a wearable RGB-D camera," *Comput. Vis. Image Und.* **154**, 192–205 (2017).
30. A. Ciobanu, A. Morar, F. Moldoveanu, L. Petrescu, O. Ferche and A. Moldoveanu, "Real-time indoor staircase detection on mobile devices," **In**: *2017 21st International Conference on Control Systems and Computer Science (CSCS)*, IEEE (2017) pp. 287–293.
31. G. C. Haynes and A. A. Rizzi, "Gaits and gait transitions for legged robots," **In**: *Proceedings of the IEEE International Conference on Robotics and Automation*, Orlando, FL, USA (May 2006) pp. 1117–1122.
32. R. Full and D. Koditschek, "Templates and anchors: neuromechanical hypotheses of legged locomotion on land," *J. Exp. Biol.* **202**, 3325–3332 (1999).
33. E. Sayginer, T. Akbey, Y. Yazicioglu and A. Saranlı, "Task oriented kinematic analysis for a legged robot with half-circular leg morphology," **In**: *IEEE International Conference on Robotics and Automation* (May 2009) pp. 4088 –4093.
34. A. M. Johnson and D. E. Koditschek, "Legged self-manipulation," *IEEE Access*, **1**, 310–334 (May 2013).
35. T. Lozano-Perez, M. T. Mason and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *Int. J. Robot. Res.* **3**(1), 3–24 (1984).
36. A. M. Johnson, G. C. Haynes and D. E. Koditschek, "Disturbance Detection, Identification, and Recovery by Gait Transition in Legged Robots," **In**: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan (October 2010).
37. J. Hesch, G. Mariottini and S. Roumeliotis, "Descending-Stair Detection, Approach, and Traversal with an Autonomous Tracked Vehicle," **In**: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (October 2010) pp. 5525–5531.
38. D. Campbell and M. Buehler, "Stair descent in the simple hexapod RHex," **In:** *IEEE International Conference on Robotics and Automation, 2003. Proceedings of ICRA'03.*, vol. 1 IEEE (2003) pp. 1380–1385.
39. A. Kalantari, E. Mihankhah and S. Moosavian, "Safe Autonomous Stair Climbing for a Tracked Mobile Robot Using a Kinematics Based Controller," **In:** *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM 2009*, IEEE (2009) pp. 1891–1896.