α-WaLTR: Adaptive Wheel-and-Leg Transformable Robot for Versatile Multiterrain Locomotion

Chuanqi Zheng, Siddharth Sane, Kangneoung Lee, Vishnu Kalyanram, and Kiju Lee, *Member, IEEE*

***Abstract*—Adaptability is a fundamental yet challenging requirement for mobile robot locomotion. This article presents α-WaLTR, a new adaptive wheel-and-leg transformable robot for versatile multiterrain mobility. The robot has four passively transformable wheels, where each wheel consists of a central gear and multiple leg segments with embedded spring suspension for vibration reduction. These wheels enable the robot to traverse various terrains, obstacles, and stairs while retaining the simplicity in primary control and operation principles of conventional wheeled robots. The chassis dimensions and the center of gravity location were determined by a multiobjective design optimization process aimed at minimizing the weight and maximizing the robot's pitch angle for obstacle climbing. Unity-based simulations guided the selection of the design variables associated with the transformable wheels. Following the design process, α-WaLTR with an embedded sensing and control system was developed. Experiments showed that the spring suspension on the wheels effectively reduced the vibrations when operated in the legged mode and verified that the robot's versatile locomotion capabilities were highly consistent with the simulations. The system-level integration with an embedded control system was demonstrated via autonomous stair detection, navigation, and climbing capabilities.**

***Index Terms*—Ground locomotion, mobile robots, multiterrain locomotion, wheel-leg transformable robots.**

I. Introduction

AUTONOMOUS mobile robots on the ground can perform various tasks, such as surveillance, transportation, search-and-rescue missions, and exploration. Common methods of terrestrial locomotion in these robots include wheels, legs, and tracks. Wheels enable the simplest yet most efficient locomotion on relatively smooth and flat surfaces, but they exhibit limitations in traversing rough terrains or obstacles. Increasing the wheel size improves overall locomotion performance; however, it also increases the size and weight of the robot, possibly limiting accessibility to confined spaces. Compared to the size of the wheels, the climbable obstacle height remains relatively small. External factors, such as the surface geometry and friction between the wheel and contact surface, also contribute significantly to the overall mobility [1], [2]. However, these external conditions can only be predictable if the robot operates in a well-known environment. Legs, on the other hand, typically outperform wheels in challenging terrain conditions, but they often suffer from mechanical complexities and control difficulties [3], [4]. Tracks or crawling mechanisms have advantages over wheels or legs on soft terrains and slopes, and their operational principles can be as simple as wheels [5], [6], [7], [8]. However, the increased mechanical complexity often limits their use to low-speed applications.

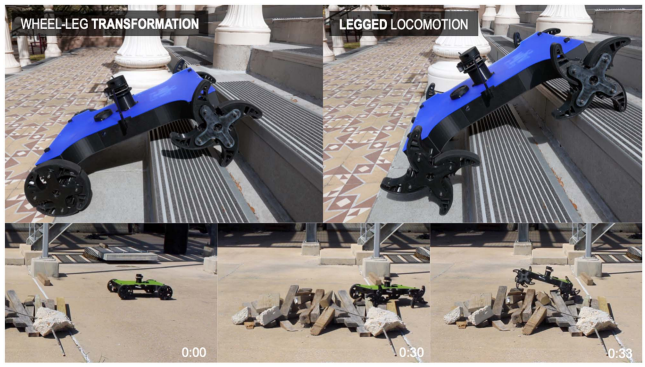


Fig. 1. α-WaLTR with passively transformable wheels for multiterrain locomotion.

A. Introducing α-WaLTR

α-WaLTR is a new adaptive wheel-and-leg transformable robot (see Fig. 1). The overall size of the platform is $0.72(L) \times 0.59(W) \times 0.30(H)$ m^3, weighing about 11 kg. The platform is equipped with an autonomous navigation system enabled by onboard sensing, processing, and wireless communication capabilities. This robot is developed for urban military operations under the support of the Defense Advanced Research Projects Agency's OFFensive Swarm-Enabled Tactics (OFFSET) program. Therefore, our technical objectives align with one of the OFFSET missions of advancing locomotion capabilities in urban military environments. However, the application domains of α-WaLTR are not limited to this. It has potential for many other areas, such as agriculture, space exploration, and surveillance.

Manuscript received 29 December 2021; revised 19 May 2022 and 3 October 2022; accepted 18 November 2022. Date of publication 14 December 2022; date of current version 5 April 2023. This work was supported by the Defense Advanced Research Projects Agency under the OFFensive Swarm-Enabled Tactics (OFFSET) under Grant HR00112020037. This paper was recommended for publication by Associate Editor C. Yang and Editor M. Yim upon evaluation of the reviewers' comments. *(Corresponding author: Kiju Lee.)*

Chuanqi Zheng, Siddharth Sane, Kangneoung Lee, and Vishnu Kalyanram are with the Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: chuanqi.zheng@tamu.edu; sidd22194@tamu.edu; eehoa12@tamu.edu; vshnuk@tamu.edu).

Kiju Lee is with the Department of Engineering Technology and Industrial Distribution and the Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: kiju.lee@tamu.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TRO.2022.3226114.

Digital Object Identifier 10.1109/TRO.2022.3226114

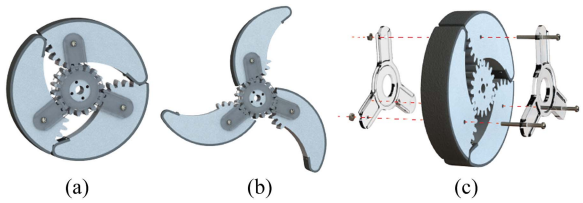


Fig. 2. WheeLeR mechanism overview: CAD assembly design of (a) closed wheel, (b) open wheel, and (c) an exploded view [9].

The locomotion system of α-WaLTR consists of four passively transformable wheels, which build on our previous design of the geared transformable wheel mechanism *WheeLeR* (see Fig. 2) [9]. This design consists of a central gear, leg segments with partial gears, and two spoke frames. When the central gear rotates in the clockwise direction, the legs can open until it reaches the mechanical locks. The entire mechanism turns and moves to the right with the legs open as it continues to rotate under sufficient friction, even at a low speed. This allows the wheel to adapt to terrain conditions. For example, the wheel remains mostly closed in a circular shape on a smooth flat surface. It transforms into the leg configuration on a rough surface or when encountering an obstacle. When the central gear rotates in the opposite direction, the legs close or remain closed in a circular shape. The free-body diagram of this wheel and the analysis of the wheel-leg transformation tendency were presented in [9]. WheeLeR is mechanically simple, with only a few moving components. The design variables, including the ratio of the partial gears on the legs to the central driving gear ($\rho = r_2/r_1$), the number of legs (n_{leg}), and the wheel radius ($R = r_1 + 2r_2$), are customizable. A small prototype ($0.115 \times 0.047 \times 0.031$ m^3; 0.5 kg) with two of these wheels and a balancing tail was built as a demonstrative proof-of-concept prototype [9].

α-WaLTR is a four-wheel drive mobile robotic platform developed for real-world tactical applications. It is much larger and heavier than our small conceptual prototype because of the necessary sensors and other hardware components required for autonomous navigation. The physical scale-up causes increased vibrations and shocks during legged locomotion, leading to an increased risk of hardware damage over time. We have now implemented a new *advanced* WheeLeR mechanism with embedded torsional springs for vibration reduction. A multidisciplinary optimization approach was adopted to determine the overall chassis dimensions and the location of the center of gravity (CG). The unique challenges associated with selecting the wheel design were addressed by creating a Unity-based simulator for comprehensive locomotion testing. α-WaLTR, with its control system based on the robot operating system (ROS), is developed and tested for versatile locomotion performance. The robot's autonomous stair detection, navigation, and climbing capabilities presented in this article demonstrate its successful system-level integration.

B. Existing Hybrid Locomotion Systems

A hybrid locomotion system aims to improve mobility by combining two or more locomotion methods, especially when the target application involves diverse and unknown terrain conditions. One way to achieve this is to equip a robot with more than one locomotion system. For example, robots with both wheels and legs can selectively use one or the other depending on the terrain type. The wheels and legs can be either separated [10] or connected through the same motor shafts [11]. Some robots have two wheels and two legs to overcome the speed limitations commonly observed in legged robots while addressing locomotion challenges in wheeled robots [12], [13]. Combining tracks and wheels can also achieve improved locomotion performance [14], [15]. Another way to achieve hybrid locomotion is to mechanically integrate more than one strategy, e.g., wheels attached at the distal tips of individual legs [16], [17], [18], [19], [20]. There are also mechanisms with legs attached to the rims of individual wheels that can be folded or stretched out [21]. *Whegs* are spoke wheels capable of generating leg-like locomotion while being operated as wheels [22]. This mechanism achieves the speed and simplicity of wheels and the versatility of legs. *TriBot*, equipped with four Whegs, is further hybridized by having an additional manipulator [23].

Transformable hybrid mechanisms allow the system to change the mode of locomotion during operation. These mechanisms can be either *active* or *passive* [9] based on their transformation strategies. Active transformation requires a dedicated actuator(s) to transition between two or more modes. In passive mechanisms, external and internal factors typically trigger transformation without involving any additional actuator. An active mechanism allows the system to use a specific locomotion method, but it increases system complexity in both hardware and software. While passive mechanisms do not increase the system complexity substantially, they introduce uncertainties in transition behavior. Most transformable mechanisms utilize the wheel-and-leg transformation due to the expected advantages and relative mechanical simplicity.

Active mechanisms have adopted several actuation strategies. One method involves multiple leg segments evenly arranged around a disc connected to an axial shaft. By pulling or pushing this disc, the leg segments can open or close [24], [25], [26], [27]. A similar mechanism was also applied to origami wheels [28], [29]. Another strategy adopts stick-shaped legs, hidden in the wheel or extended out [30], [31]. A wheel comprised of two half-circle legs can be either folded into a semicircular wheel [32], [33] or deviated radially, forming two legs [34], [35]. A commonly adopted three- or four-leg wheel design consists of arciform lobes connected through linkages that cause the lobes to fold or extend [36], [37], [38], [39]. Deformable wheels presented in [40], [41], and [42] could switch between circle and ellipse and also use the track motion. While most of the existing active mechanisms use a motor for transformation, some use wire-driven actuation for its simplicity and lightweight [43], [44]. Many active transformable mechanisms contain an actuator within the wheel, requiring power and control signal transmission from the main chassis; several previous studies used a slip-ring device for this [45], [46], [47].

Compared to active mechanisms, relatively fewer passive ones exist. *Wheel Transformer* consists of two normal legs and one triggering leg assembled with a transmitting disc and a spoke frame [49]. The legs can open passively when an external

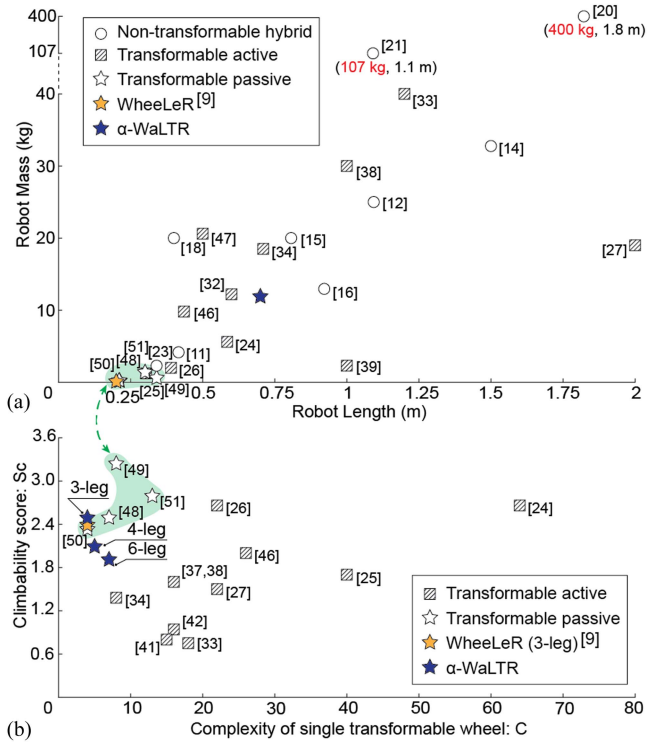


Fig. 3. Comparisons with the existing hybrid mechanisms. (a) Weight versus size of the mobile robots with hybrid locomotion systems. (b) S_c versus C of transformable wheel-leg mechanisms.

frictional force acts on the triggering leg. The passive morphing wheel in [50] consisted of a frame and three leg segments. In this design, springs and magnets keep the legs closed when the robot stops or moves at a low speed, and wheel-to-leg transformation is triggered when the robot drives at a high speed. Another existing design consists of three leg segments, three links, an internal spoke frame, and an external spoke frame [51]. The leg segments are connected to the internal spoke frame directly and to the external spoke frame through the links. Transformation is triggered by external surface conditions, such as the existence of an obstacle. Another design consists of three scissor chain legs and two discs in the middle, forming a rotating pair. The mechanism is passively transformable by connecting one side of the scissor chain to a leg and the other side to the inner disc [52].

C. Comparisons With the Existing Hybrid Mechanisms

Based on our review of the existing platforms, passive transformation in hybrid mechanisms has not been realized in physical platforms of the scale of α-WaLTR to date. Fig. 3(a) compares some existing hybrid locomotion systems and our previous WheeLeR prototype with α-WaLTR in terms of the weight and the size represented by the longest body length. While nontransformable hybrid and actively transformable robots have been developed in varying sizes and weights, passively transformable ones have remained relatively small and light. Passive mechanisms often rely on high surface friction or a high rotating speed (or both) for triggering wheel-to-leg transformation [49], [50]. Such transformation strategies may be suitable only for small lightweight robots. In WheeLeR, the leg segments are mechanically connected to the driving system through the gears. Therefore, the driving actuator and external factors (e.g., surface friction and obstacle) trigger the transformation even under a low speed and high payload.

For the wheel-leg transformable mechanisms, we further evaluated the robot's overall obstacle-climbing ability and structural complexity. We define a measure of the obstacle climbing ability as

$$\mathcal{S}_c = H_o^{\max}/R \tag{1}$$

where $H_o^{\max}$ is the maximum climbable obstacle height and R is the closed wheel radius. Mechanical complexity is measured by the metric defined in [53], i.e., $C = N_A \cdot N_J$, where N_A is the number of actuators and N_J is the number of joints. As a robot may have a different number of wheels, C is calculated for a single wheel. A regular wheel driven by a motor has one actuator and one joint, resulting in $C = 1$. C increases as the design becomes mechanically more complex. Fig. 3(b) shows $\mathcal{S}_c$ versus C for some existing wheel-leg transformable mechanisms. While our transformable mechanism can have any number of legs, the most practical choices are $n_{\text{leg}} =$ 3-6. Fig. 3(b) shows α-WaLTR equipped with three-, four-, and six-leg versions of the wheels that were fabricated for physical experiments. We can modify the design to increase the $\mathcal{S}_c$ values by extending the leg length, as described in Section II-A. The three-leg design achieves the lowest C with a relatively high $\mathcal{S}_c$ compared to the four- or six-leg versions as well as other transformable wheel-leg mechanisms shown in the plot. In general, passive mechanisms achieve relatively high $\mathcal{S}_c$ with a low C. Most of the existing robots with passive mechanisms are small and lightweight, occupying the lower left side of the plot in Fig. 3(a). α-WaLTR is significantly larger and heavier than those while maintaining the mechanical simplicity and climbing ability expected in the passive systems.

II. Advanced WheeLeR Mechanism

The kinematic model of the transformable wheel mechanism was analyzed and reengineered to be physically *scalable* and *modular*. The new WheeLeR design with embedded torsional springs was introduced for vibration reduction when used in a larger and heavier platform.

A. Kinematic Model and Analysis

The kinematic analysis aimed to 1) determine the relationship between the rotating angle of the central gear (θ_c) and the distance between the wheel center and the leg tip (R_{tip}) and 2) analyze the vertical trajectories of the mechanism in the legged configuration. The length of the legs and the opening angle determine the maximum value of R_{tip}, which directly influences the climbing ability of the mechanism. When the wheel is in the legged configuration, its "walking" motion generates vertical movements when the leg segments alternatively touch the ground while rotating. We expect passive wheel-to-leg transformation to occur when the wheel encounters an obstacle

TABLE I
VARIABLES IN THE KINEMATIC MODEL

Variable	Description
n_{leg}	Number of legs
r_1	Radius of the central gear
r_2	Radius of the partial gear on the legs
ρ	Gear ratio of r_2 to r_1 $(= r_2/r_1)$
R	Radius of the closed wheel $(= r_1 + 2r_2)$
r_{tip}	Radius of the rounded leg tip
θ_c	Rotating angle of the central gear
θ_{leg}	Rotating angle of the leg
R_{tip}	Distance between the wheel center (O) and the leg tip (B)

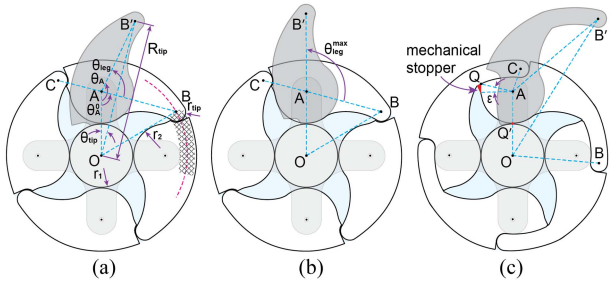


Fig. 4. Kinematic model of the transformable wheel.

or operates on rough terrain. However, the projected moving trajectories of the legged configuration on a flat surface provide useful information for understanding the relationship between the WheeLeR design and vertical motion. Table I defines variables introduced in the kinematic model and analysis.

Fig. 4 shows the kinematic model of WheeLeR. **O** denotes the wheel center, **A** is the center of gear on the leg, and **B** is the center of the rounded tip of the leg. **B** in the closed wheel configuration can be located along the dashed arc overlapping with the adjacent leg segment. When **B** is located further away from **A**, it creates a longer leg and, thus, is likely more suited for climbing a higher obstacle. However, it requires a higher torque at **O** with a longer moment arm. For WheeLeR to form a circular shape when fully closed in order to roll smoothly on a flat surface, the gaps between the leg tip and the rear end of the adjacent leg segment must be small. We designed the rear side to match the geometry of the leg tip to ensure minimal gap, as shown in Fig. 4.

Given n_{leg}, r_1, r_2, r_{tip}, and selected location of **B**, $\overline{AB}$, $\overline{AC}$, $\angle BOA$, $\angle AOC$, $\angle BAC$, and $\angle OAB$ in the closed wheel configuration can be obtained. We know $\angle BOC = 2\pi/n$ and denote $\angle OAB$ as θ_A^0. The total rotating angle of the leg segment ($\angle OAB'$) becomes

$$\theta_A = \theta_A^0 + \theta_{\text{leg}} = \theta_A^0 + \theta_c \frac{r_1}{r_2} \tag{2}$$

where $\theta_0^A \le \theta_A \le \theta_A^{\max}$.$R_{\text{tip}}$ is measured by the distance from **O** to **B**′, calculated as

$$R_{\text{tip}} = \sqrt{\overline{OA}^2 + \overline{AB}^2 - 2\overline{OA}\cdot\overline{AB}\cos\theta_A} \tag{3}$$

where $\overline{OA} = r_1 + r_2$. Based on the geometric relationships shown in Fig. 4(b), $R_{\text{tip}}^{\max} = \overline{OA} + \overline{AB}$ is achieved when $\theta_A^{\max} = \pi$. However, this may not be achievable depending on the selected design variables and the location of **B**. The example shown in Fig. 4(c) has $\theta_A^{\max} < \pi$. **Q** is located where the end of the partial gear on the rear side of the leg meets the inner surface of the adjacent leg. Then, $\angle QAO$ is calculated by

$$\angle QAO = \cos^{-1}\left\{\frac{\overline{AQ}^2 + \overline{AO}^2 - \overline{QO}^2}{2\overline{AQ}\cdot\overline{AO}}\right\}$$

where $\overline{AQ} = r_2$, $\overline{AO} = r_1 + r_2$, and $\overline{QO} = R - 2r_{\text{tip}}$. $\angle QAO$ is the total angle that the leg can rotate about the central gear. However, a mechanical stopper must be added to prevent the leg from further rotation. Then, $\theta_{\text{leg}}^{\max}$ is determined by

$$\theta_{\text{leg}}^{\max} = \angle QAO - \epsilon \tag{4}$$

where ϵ is the reduced angle due to the added stopper design.

We adopted four different designs of WheeLeR and analyzed the vertical trajectories of their legged configurations on a flat surface. The first three designs used the same $r_1 = r_2 = 0.032$ with different numbers of legs, $n_{\text{leg}} = 3$, 4, and 5, respectively. The fourth design used $r_1 = 0.048$, $r_2 = 0.024$, and $n_{\text{leg}} = 6$. The location of **B** in each design is arbitrarily selected within the reasonable range. Fig. 5 shows the four designs and the corresponding vertical trajectories of **O** measured from the ground ($\overline{OG}$) as the wheel rotates to the right in the legged configuration.

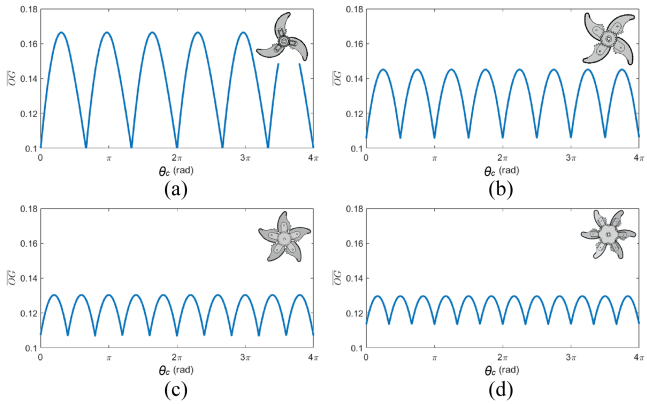


Fig. 5. Vertical trajectories of the wheel in four different designs. (a) $n_{\text{leg}} = 3$ and $r_1 = r_2 = 0.032$. (b) $n_{\text{leg}} = 4$ and $r_1 = r_2 = 0.032$. (c) $n_{\text{leg}} = 5$ and $r_1 = r_2 = 0.032$. (d) $n_{\text{leg}} = 6$ and $r_1 = 2r_2 = 0.048$.

B. Advanced WheeLeR Design With Spring Suspension

In a small-size lightweight robot, the impact force directly applied to the motor shaft and vibrations experienced by the whole body are insignificant. However, a larger (and heavier) robot equipped with these wheels would continuously experience increased shocks and unwanted vibrations while operating in the legged mode, as observed in Fig. 5. Such shocks and vibrations would not only increase wear on the structure but also affect the sensor readings, causing control difficulties. Owing to the impact of these shocks and vibrations, they must be reduced. Modularity is another important design consideration, as on-site replacement of the wheels is often required in field operations.

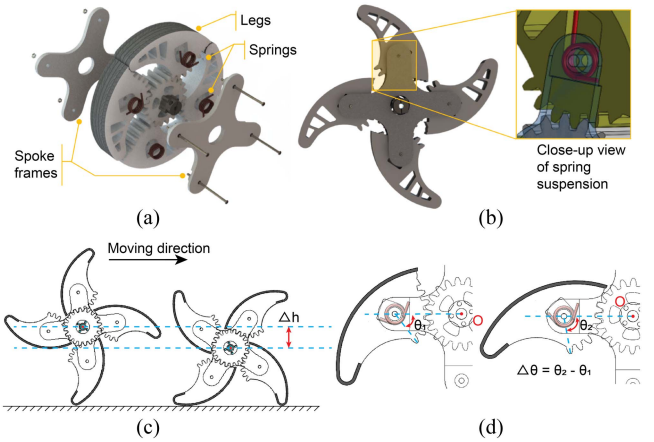


Fig. 6. Advanced WheeLeR design with embedded torsional springs. (a) Exploded view of the CAD model. (b) Fully assembled view with a close-up view of the torsional spring housed in the leg. (c) Vertical height change while walking. (d) Winding angle in the torsional springs.

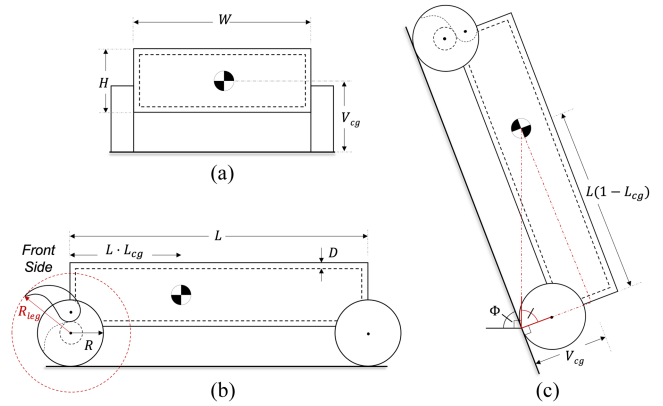


Fig. 7. Visualization of seven design variables ($L, W, H, D, L_{\text{cg}}, V_{\text{cg}}$, and R) subject to optimization and the extended wheel radius when leg segments are fully open (R_{leg}) (a) and (b), and maximum pitch angle (Φ) (c). (a) Front view. (b) Side view. (c) Maximum pitch angle.

The advanced WheeLeR design for α-WaLTR features an embedded torsional spring added to each leg assembly, as shown in Fig. 6(a). As it rotates in the legged mode, the torsional springs reduce shocks. Each leg on the wheel is equipped with a cavity to install a right-angled torsional spring. One end of the spring is fixed in the cavity on the leg segment, while the other is left free [see Fig. 6(b)]. As one leg hits an obstacle or the ground while walking, the free ends of the springs hit the spoke frame, winding the springs and absorbing the impact. This design is fully integrated, making the wheel structure highly modular while maintaining its mechanical simplicity.

C. Selection of Torsional Springs

The selection of torsional springs in the advanced WheeLeR mechanism requires careful consideration of both the physical space and the expected torque acting on each spring while walking. The target amount of energy to be absorbed by the springs is estimated to be the total change in potential energy due to the vertical height change while walking (Δh) [see Fig. 6(c)]. This leads to an approximation of $\mathcal{M}_r g \Delta h \approx 4n_{\text{leg}}(1/2k\Delta\theta^2)$, where $\mathcal{M}_r$ is the total mass of the robot, g is gravitational acceleration, Δh is the maximum change in the vertical height of the robot body while walking, k is the spring constant, and $\Delta\theta$ is the range of winding angle of the torsional spring [see Fig. 6(d)]. The right side of the formula is multiplied by 4 because our platform is equipped with four wheels. The design of the spring housing in the leg segment can be modified to adjust $\Delta\theta$. Considering practical and space limitations, its range is kept at $45^\circ \leq \Delta\theta \leq 50^\circ$ in our designs.

The stiffness and the winding angle can, then, be used to determine the strain energy stored in each spring due to the bending moment. A spring grade is, then, determined depending on the incident stress [54]. In this step, the mean diameter and the wire diameter are first selected based on the physical space constraints. Subsequently, the yield strength of the spring and the spring index are obtained. The Wahl factor is used to calculate the bending stress on the spring. If the obtained bending stress is less than the ultimate tensile strength of the spring, the design parameters are considered to be safe. Otherwise, if higher values for the mean and wire diameter are chosen, the process continues until the design is safe. This process must also ensure that the natural frequency of the spring would not result in resonance. The total weight and the mechanical design and size of the wheels must be determined first in order to select proper springs.

III. Multidisciplinary Design Optimization

Designing α-WaLTR involved many variables and parameters to be carefully identified and examined. Optimizing the design variables for desired locomotion performance can be challenging and often time consuming. This section describes the multidisciplinary system analysis and design optimization adopted for selecting upper level design variables, including the overall dimensions of the chassis, location of the CG, and the wheel size.

A. Design Variables, Parameters, and Constraints

We assume that the robot chassis is in a simple rectangular box shape and wheels are in a disc shape. Design variables include the chassis length (L), width (W), height (H), thickness of the chassis wall (D), relative longitudinal position of the CG measured from the front end of the chassis (L_{cg}), vertical position of the CG (V_{cg}), and closed wheel radius (R). Seven variables subject to optimization are illustrated in Fig. 7(a) and (b). The moving direction where the wheels can transform into legs is considered *forward* and that side of the chassis is referred to as the *front*. L_{cg} ranges between 0 and 1 and is 0.5 if the CG is at the center of the body chassis. A value smaller than 0.5 indicates that the CG is closer to the front end. We assume that the front and rear wheels are mounted at each end of the body chassis. Only R is considered among the wheel design variables at this optimization phase. Parameters include the payload (P), the wheel width (w), and the estimated average densities of the materials used for the chassis (d_c) and the wheels (d_w).

TABLE II
LIST OF DESIGN VARIABLES AND PARAMETERS WITH THEIR BOUNDARY CONDITIONS (AND SELECTED VALUES) AND JUSTIFICATION

Variable	Description	Boundary Conditions [meters(m)]	Justification
L	Chassis length	$0.5 \leq L \leq 0.7$	Fitting all the onboard components without exceeding 0.7m
W	Chassis width	$0.3 \leq W \leq 0.7$	Fitting all the onboard components without exceeding 0.7m
H	Chassis height	$0.08 \leq H \leq 0.1$	Fitting the tallest onboard component with some extra space
D	Chassis wall thickness	$0.005 \leq D \leq 0.01$	Reasonable range without possible bending/warping
L_{cg}	Relative longitudinal CG position	$0.25 \leq L_{cg} \leq 0.5$	Based on the CG analysis
V_{cg}	Vertical position of the CG	$0.16 \leq V_{cg} \leq 0.2$	Based on the CG analysis
R	Closed wheel radius	$0.095 \leq R \leq 0.12$	Based on traversable gap, obstacle height, and V_{cg}
Parameter	**Description**	**Value**	**Justification**
w	Wheel width	$0.03\ m$	Selecting a reasonable arbitrary value
d_c	Chassis material density	$1,250\ kg/m^3$	Considering potential chassis materials
d_w	Wheel material density	$500\ kg/m^3$	Considering potential wheel materials
P	Internal payload	$6\ kg$	Estimated total weight for embedded hardware components

Boundary values of L and W were determined by the minimum required area for fitting all the internal components without exceeding 0.7 m along each dimension. The upper boundary was selected by the size of the larger commercial unmanned ground vehicle platforms used by the OFFSET program. The lower boundary of H was determined by the tallest internal component (i.e., Jetson TX2) plus adding some room for heat dissipation while keeping the maximum height within 0.1 m. The boundaries of L_{cg} considered possible arrangement of the internals while keeping it toward the front. V_{cg} is the vertical position of the CG measured from the ground, and its boundaries were determined by the reasonable wheel and chassis sizes. The standard raise height of a stair is around 0.18 m, which was used as the minimum climbable obstacle height. Using the same value for the maximum traversable gap width, the lower bound of the closed wheel radius $R = 0.095$ m was determined. The upper boundary considers low V_{cg} as well as the overall size.

Table II lists all the design variables with the boundary conditions and parameters with selected values used for this optimization process. Parameter values used in the optimization were selected based on reasonable estimations. The wheel width does not directly affect the climbing capability, and a fixed value of $w = 0.03$ m was selected. Material densities are determined based on commonly used materials for robot chassis and wheels. The wheels were 3-D-printed using polylactic acid (PLA) with a material density 1250 kg/m^3. With 40% infill percentage, $d_w =$ 500 kg/m^3. We used $d_c = 1250$ kg/m^3 considering potential candidates of the chassis materials (e.g., carbon fiber, PLA, and acrylics) for the chassis. Since α-WaLTR was developed to work with other robotic agents in the swarm as part of the OFFSET program, it must carry specific sensing and processing devices as well as the actuators and batteries, as listed in Section V-A. The total estimated weight of those components are $P = 6$ kg. This only considers the internal payload associated with these embedded components, not the external payload to be carried by the robot for transportation. α-WaLTR was not designed to carry external payload, and thus, we used a fixed parameter value for P. For a robot carrying an external payload, an estimated maximum value may be added to the internal payload.

A few assumptions also had to be made regarding the constraints. The chassis base could sag over time due to the load on the chassis and the impact it would experience while moving on rough terrain or in legged locomotion. Sagging will affect the climbing capability and result in misaligned motors. We assume that less than a millimeter of sagging may happen. We also keep the ratio between the wheelbase and the track between 1.2 and 1.6 based on common vehicle building standards. Finally, the vehicle may topple on its side while taking a sharp turn or turning on an inclined surface. The static stability factor (SSF) was calculated by the ratio between the half-length of the wheel track to V_{cg}, such that SSF $= \mathcal{W}/2V_{\text{cg}}$). We assume that the SSF is kept between 1.1 and 1.5.

B. Multiobjective Design Optimization

This optimization aims to determine the most advantageous set of variables under two primary objectives: 1) minimizing the robot's mass and 2) maximizing the pitch angle of the robot. The robot's mass is directly linked to torque requirements, deployability, and also the cost. The pitch angle indicates the achievable slope of the robot body without flipping over.

The first objective function for the mass is defined as

$$\mathcal{Z}_m = \frac{f_m}{z_m^{\max}} \left(P + \mathcal{M}_w + \mathcal{M}_c\right) \tag{5}$$

where P is the payload and $\mathcal{M}_w$ and $\mathcal{M}_c$ are the wheel and chassis masses, respectively, estimated by

$$\mathcal{M}_w = 2d_w \pi w R^2 \tag{6}$$

$$\mathcal{M}_c = d_c\{LWH - (L-2D)(W-2D)(H-2D)\}. \tag{7}$$

We note that $\mathcal{M}_w$ estimates the mass of two wheels only, assuming that two rear wheels remain in contact with the ground when climbing. $\mathcal{M}_c$ is determined by the volume of the box-shaped chassis multiplied by the material density d_c. f_m is the relative weight factor assigned to this objective function, and its value depends on the designer's choice. While keeping the summation of the two weight factors equal to 1, a higher weight can be assigned to the more critical objective function. If all the objectives are equally important, the same weight can be used without affecting the optimization process. $z_m^{\max}$ is the maximum possible robot mass calculated by maximizing the mass term $(P + \mathcal{M}_w + \mathcal{M}_c)$ in (5) using the sequential quadratic programming algorithm for optimization in MATLAB [55]. Based on the boundary conditions and parameter values, the result was 45 kg. This value substitutes $z_m^{\max}$ and transfers the objective function to a dimensionless space. The

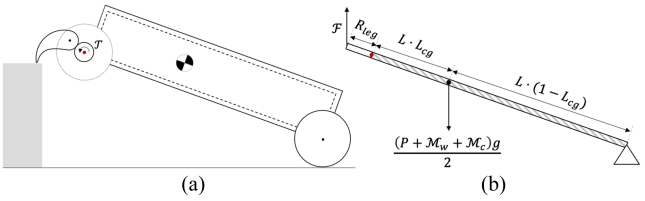


Fig. 8. (a) and (b) Simplified representation of the robot supported by the two front legs on an obstacle, while the two rear wheels remain in contact with the ground.

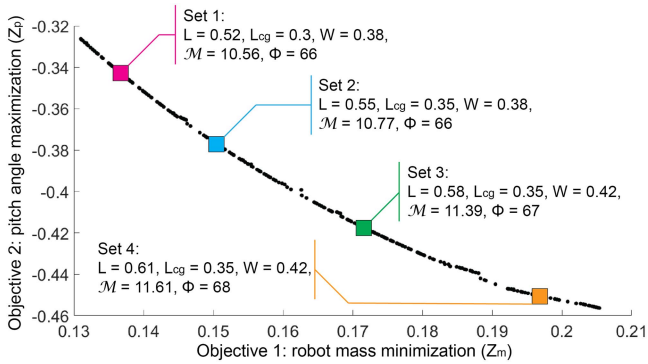


Fig. 9. Optimized objective space by the Pareto front. Four selected optimum points and corresponding variables are displayed (Set 1–4). All lengths are shown in meters except for L_{cg}, which is a fraction, the total robot weight $\mathcal{M}$ in kilograms, and the pitch angle in degrees.

two objectives involved in our optimization process do not have the same engineering units, and this transfer process allows us to evaluate the tradeoffs between the two objectives.

The driving motors' torques required for the robot to climb over an obstacle is proportional to $\mathcal{Z}_m$. When the robot is supported by the tips of the two front legs on an obstacle [see Fig. 8(a)], the torque on each wheel can be estimated based on a simplified representation shown in Fig. 8(b). When stationary, the normal force at the tip of each leg ($\mathcal{F}$), assuming that the two rear wheels remain in contact with the ground, can be approximated as

$$\mathcal{F} \approx \frac{(P + \mathcal{M}_w + \mathcal{M}_c)g}{2} \cdot \frac{L(1 - L_{\text{cg}})}{(R_{\text{leg}} + L)} \propto \mathcal{Z}_m \tag{8}$$

where $R_{\text{leg}} = R_{\text{tip}} + r_{\text{tip}}$ and R_{tip} is calculated using (3). The torque at the motor can, then, be estimated by

$$\mathcal{T} \approx \mathcal{F} \cdot R_{\text{leg}} \propto \mathcal{Z}_m. \tag{9}$$

Since the torque is proportional to the robot's mass, it is minimized when $\mathcal{Z}_m$ is minimized.

Fig. 7(c) illustrates how the maximum pitch angle (Φ) can be calculated. The second objective function for the robot's pitch angle is, then, defined as

$$\mathcal{Z}_p = \frac{f_p}{z_p^{\max}} \tan^{-1}\left\{\frac{L \cdot (1 - L_{\text{cg}})}{V_{\text{cg}}}\right\}. \tag{10}$$

Like f_m, the weight factor f_p is the designer's choice and is based on optimization goals. $z_p^{\max}$ is the maximum possible value of Φ, calculated as 74° by following the same process described above for $z_m^{\max}$ [55].

The combined objective function is defined as

$$\mathcal{Z} = \mathcal{Z}_m - \mathcal{Z}_p. \tag{11}$$

Since our goal is to minimize $\mathcal{Z}_m$ and maximize $\mathcal{Z}_p$, the negative sign was added to $\mathcal{Z}_p$ to make it a minimization problem. As previously mentioned, f_m and f_z are the weights assigned to the objective functions in (5) and (10), respectively, by the designer based on their relative importance. We used $f_m = 0.45$ and $f_p = 0.55$, assigning a slightly higher weight to the pitch angle than the total mass. Based on the described assumptions and constraints, we used the Pareto Search Algorithm in MATLAB to locate nondominated points iteratively in the objective space to generate a Pareto Front via pattern search [55]. The Pareto Front in Fig. 9 plots 200 optimal points while informing the tradeoffs between the two objectives. Four optimal points are selected to display the corresponding values of the optimized variables. For each selected point, the figure provides the corresponding total mass of the robot with four wheels ($\mathcal{M} = P + 2\mathcal{M}_w + \mathcal{M}_c$) and pitch angle ($\Phi$). Four of the seven variables always converge to the lower bounds (i.e., $R = 0.095$ m, $D = 0.005$ m, $H = 0.08$ m, and $V_{\text{cg}} = 0.16$ m) and, thus, are omitted in the figure. We note that the robot's total mass, here, accounts for the payload, the chassis mass, and the mass of four wheels instead of two. In our case, achieving a high pitch angle for obstacle climbing was considered more important than reducing the weight; therefore, Set 4 was selected for further simulations and hardware construction. If a smaller Φ with a lower $\mathcal{M}$ is desired, one may consider Set 1 instead.

IV. Locomotion Simulator

A Unity-based simulator was created to empirically evaluate the locomotive performance of various wheel designs and sizes. The design optimization process described above suggests optimal values for chassis dimensions and, thus, guides motor selection given physical constraints, payload, and minimum obstacle height. However, the performance and behavior of the passively transformable wheels on different terrains is hard to predict. Wheel-specific design variables, such as R, ρ, and n_{leg}, affect locomotive performance. The examination of these variables via physical prototyping followed by experiments is highly time consuming and costly. The presented simulator allows comparative evaluations of various wheel design options on diverse test environments.

A. Simulation Environment

Unity offers an easy user interface and a rich integrated development environment for robot simulations [56]. For example, Unity-based simulators have been linked with the ROS to develop and test navigation and control algorithms of unmanned aerial or ground vehicles [57], [58] as well as multirobot systems [59]. The virtual locomotion test environment and robot models were created in Unity 2018.4.12f1 (Unity Technologies

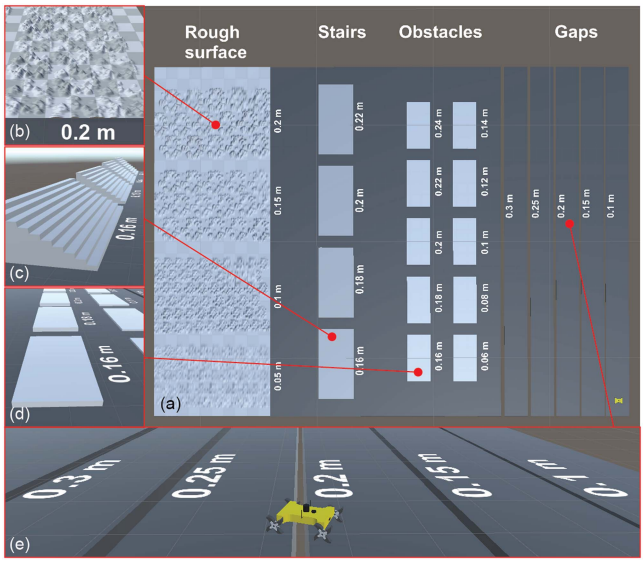


Fig. 10. Unity simulation environment consisting of multiple terrains, including rough surfaces, stairs, obstacles, and gaps.

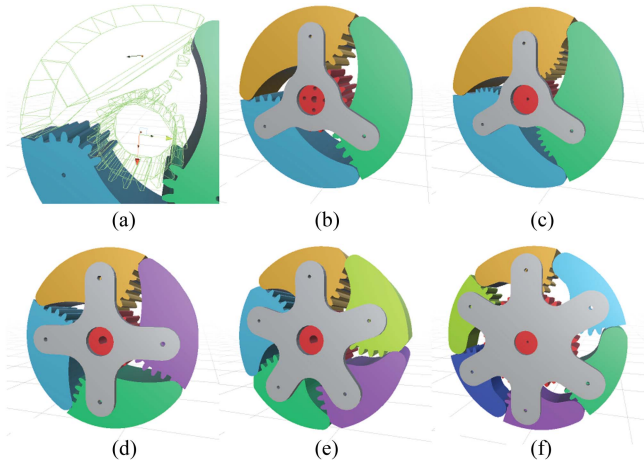


Fig. 11. Unity models for WheeLeR with different design parameters. (a) Colliders of central gear and leg segment. (b) Design I: $n_{\text{leg}} = 3$ and $\rho = 1$. (c) Design II: $n_{\text{leg}} = 3$ and $\rho = 2$. (d) Design III: $n_{\text{leg}} = 4$ and $\rho = 1$, (e) Design IV: $n_{\text{leg}} = 5$ and $\rho = 1$. (f) Design V: $n_{\text{leg}} = 6$ and $\rho = 0.5$.

Inc.) on a Windows 10 computer with the following system configuration: Intel Core$^{\text{TM}}$ i7-8700K CPU @ 3.70 GHz; 32.0-GB DDR4 RAM @ 2666 MHz; NVIDIA GeForce GTX 1080 Ti; 256-GB M.2 PCIe NVMe SSD.

B. Modeling of Test Environment

Locomotion performance may vary significantly across different situations and projected environmental conditions. Most real-world applications involve a significant level of uncertainty and diverse terrain conditions, making the robot's versatile locomotion capability critical. Benchmarking the experimental protocol presented in [53], a modular set of virtual environmental structures was created in Unity, as shown in Fig. 10. While these structures are highly modular and customizable, the current test setup consists of the following:

1) *gaps* with varying widths of 0.1, 0.15, 0.2, 0.25, and 0.3 m;
2) *obstacles* with varying heights 0.06, 0.08, . . ., 0.24 m;
3) *stairs* with the tread of 0.25 (standard depth) and varying raise height of 0.16, 0.18, 0.20, and 0.22 m;
4) *rough surfaces* with irregular bumps with varying average heights of 0.05, 0.1, 0.15, and 0.2 m.

C. Modeling of α-WaLTR

Creating a Unity model of α-WaLTR involves the following four steps: 1) importing 3-D computer-aided design (CAD) models of the wheel components to Unity; 2) creating colliders, which define the physical shapes and boundaries of objects, for individual moving components of the wheel; 3) assembling the wheel components; and 4) connecting four wheels to a robot chassis model. First, SolidWorks 3-D models of the transformable wheels are created and imported to Unity. Second, colliders are defined for individual moving components, including the central gear and leg segments, as shown in Fig. 11(a). For wheel assembly, all the joints connecting the gears and legs to the spoke frame were defined as configurable joints, which provide customizability and guarantee the accuracy of the shaft positions. Finally, four wheels are assembled to the chassis. The hinge joints connect the central gears to the chassis, and each joint is integrated with a simulated motor. We used 18 N · m for the simulated motor torque, matching the motor output in the physical robot. A simple P control is applied to the motors with a target rotational velocity specified. The chassis dimensions followed the suggested values in Set 4 (see Fig. 9).

The gear ratio ρ affects the wheel-to-leg transformation tendency, i.e., the larger value of ρ, the easier transformation from wheel to legs [9]. n_{leg} also influences the locomotive behavior, such that the wheel with a higher n_{leg} would result in smoother walking, while $\mathcal{S}_c$ (1) becomes smaller than that with a smaller n_{leg}. For empirical evaluations of locomotion performance, five wheel designs were created as shown in Fig. 11(b)–(f): [I] $n_{\text{leg}} = 3$ with $\rho = 1$, [II] $n_{\text{leg}} = 3$ with $\rho = 2$, [III] $n_{\text{leg}} = 4$ with $\rho = 1$, [IV] $n_{\text{leg}} = 5$ with $\rho = 1$, and [V] $n_{\text{leg}} = 6$ with $\rho = 0.5$. For each design option, we considered varying wheel sizes of $R = 0.08 - 0.11$ with a 0.005-m increment, resulting in 35 sets of wheels for evaluation. While $R = 0.095$ m satisfies the minimum traversable gap, we opted to explore varying wheel sizes in simulations for comprehensive understanding of the dynamic behavior and relationships between the wheel design, size, and locomotion performance.

D. Simulation Protocols and Results

We conducted simulations to compare the locomotion performance of a robot equipped with conventional wheels and different designs of WheeLeR in varying sizes and operated at different speeds. The chassis model based on Set 4 was first equipped with regular (nontransformable) wheels in varying sizes of $R = 80$–110 mm with a 5-mm increment and tested on diverse terrains, including stairs, obstacles, gaps, and rough surfaces, as shown in Fig. 10. Then, the robot model was

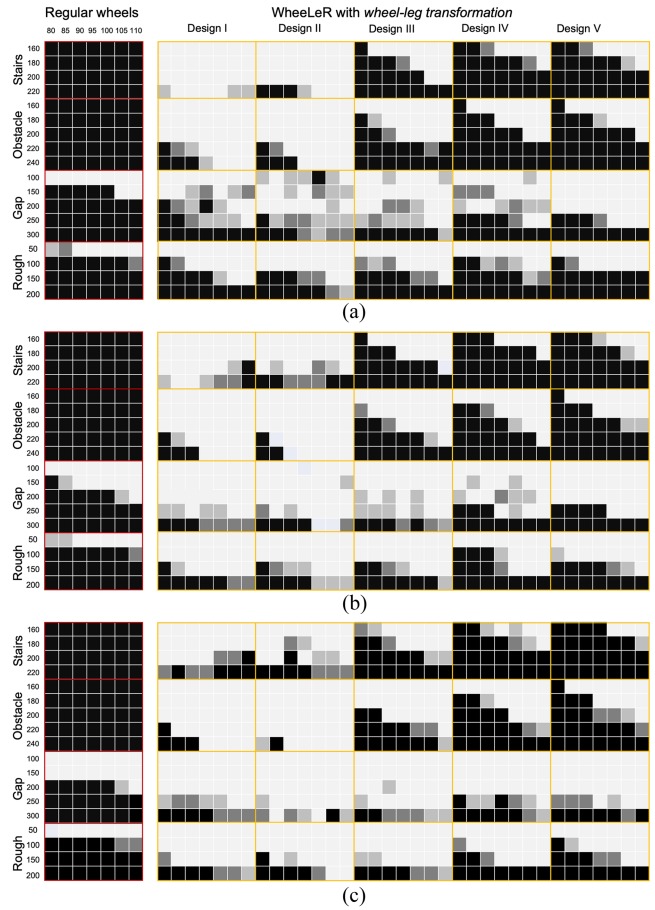


Fig. 12. Simulation results of locomotion testing at two different speeds, (a) 1.6, (b) 3.2, and (c) 4.5 rad/s. Regular wheels and five designs (I–V) of WheeLeR with $R = 0.08 : 0.005 : 0.11$ were tested on four different types of environment of stairs, obstacles, gaps, and rough terrains. 3/3 success trials shown in white, 2/3 in light gray, 1/3 in dark gray, and 0/3 in black.

equipped with WheeLeR in one of the five different designs and tested on the same set of terrains. For each design of WheeLeR, R = 80-110 with a 5-mm increment was also considered. We programmed the robot with each set of wheels to move forward on a given surface three times at the wheel speeds of 1.6, 3.2, and 4.5 rad/s and counted the number of successes. If the robot traverses the terrain within the specified time (i.e., 40 s for staircases and rough surfaces; 10 s for obstacles and gaps), we counted it as a successful trial.

Fig. 12 shows the simulation results for each speed option, where the brightest cells indicate 3 out of 3 successful trials, light gray for 2/3, dark gray for 1/3, and black for 0/3 success rate. The cells on the left with red outlines show the results from the regular wheels. The results from the five WheeLeR designs are on the right with yellow outlines. Overall, wheel-leg transformation in WheeLeR significantly enhances the overall locomotion performance of the robot compared to the regular wheels. Comparing the five different WheeLeR designs within each row, the brighter regions tend to decrease as the number of legs increases on stairs and obstacles. The differences when testing on surfaces with gaps and rough terrains were not as significant. We also observed the improved overall locomotion performance as the speed increased, except for the stairs. The stair climbing performance worsened slightly as the speed increased. The high speed occasionally caused the robot to flip over while attempting to traverse a staircase, especially for Designs I and II. The results indicate a better overall locomotion ability with a smaller n_{leg} (indicating a longer R_{tip}) and a larger R. The overall climbing score $\mathcal{S}_c$ decreases as n_{leg} increases (i.e., $\mathcal{S}_c = 2.5$ in Design I, 2.1 in Design III, and 1.92 in Design V with $R = 0.095$).

The robot's turning performance was tested by rotating the wheels in the clockwise direction on asphalt, concrete, tiles, and rough surfaces (0.02 and 0.05 m). Asphalt, concrete, and tiled surfaces were created with different dynamic friction factors (0.68, 0.80, and 0.40) and static friction factors (0.68, 1.00, and 0.40) on a flat surface. Since α-WaLTR can turn on the spot, we omitted the experiments for turning with a radius. The following scoring method is adopted for measuring turning performance: $\mathcal{S}_t = T_t^{\min}/T_t$, where T_t is the time it takes for a full 360° turn and $T_t^{\min}$ is the minimum turning time given the angular velocity of the wheels $\omega^{\max}$ assuming the wheels move on a complete circular path, calculated by $T_t^{\min} = W\pi R\omega^{\max}$.

The legs may open even on a flat surface if the friction is sufficiently high. To evaluate the walking behavior of different wheel designs on a flat surface, we tested the five designs with a fixed wheel size $R = 0.095$. Standard deviations of the vertical trajectories of V_{cg} were 0.149 for Design I, 0.133 for Design II, 0.044 for Design III, 0.038 for Design IV, and 0.031 for Design V. A smaller n_{leg} causes higher fluctuations in the vertical motions, which are directly linked to the physical vibrations and shocks experienced by the robot while walking. The standard deviation significantly reduces when $n_{\text{leg}} \geq 4$ (Designs III–V) compared with $n_{\text{leg}} = 3$ (Designs I and II).

Based on these results, Design III with $R = 0.095$ or larger satisfies our locomotion objectives. This also aligns with the results from the multiobjective optimization. For this selected design, turning performance was also tested for two sizes of $R = 0.095$ and $R = 0.11$. The results showed that $S_t = 0.65$ with $R = 0.095$ and 0.64 with $R = 0.11$ on concrete, 0.63 and 0.66 on asphalt, 0.49 and 0.53 on a rough surface with $h = 0.02$, and 0.45 and 0.45 on a rough surface with $h = 0.05$ for the two wheels, respectively.

V. Hardware Development

The Set 4 variables (see Fig. 9) for the chassis and Design III in two different sizes ($R = 0.095$ and 0.11) for the wheels are used for hardware construction. For further physical evaluations of locomotion performance and validation of the Unity simulator, two additional sets of wheels, including Design I with $R = 0.08$ and Design V with $R = 0.11$, were also fabricated. The overall software architecture is based on the ROS.

A. Embedded Hardware Components

Fig. 13 shows the embedded electronic components and actuators in a wiring diagram. Their technical specifications are as follows.

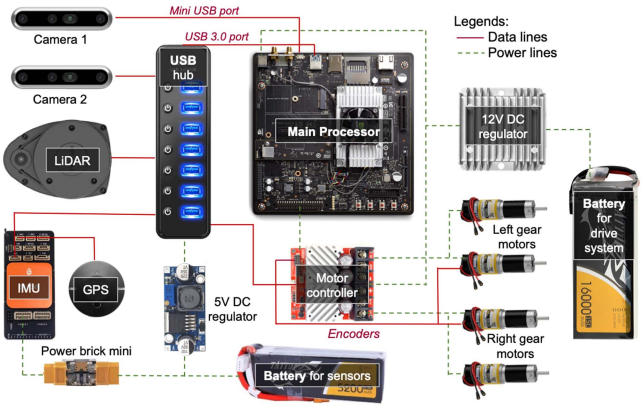


Fig. 13. Wiring diagram of all embedded components of α-WaLTR.

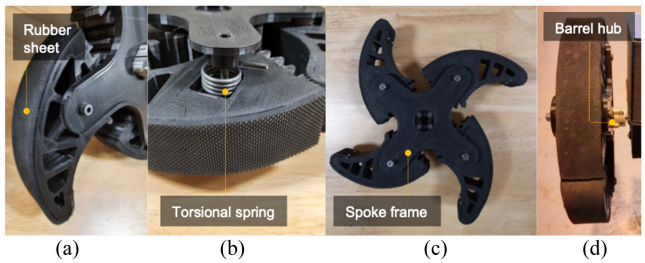


Fig. 14. Fabrication of advanced WheeLeR in four-leg configuration with spring suspension. (a) Rubber sheet added to the leg. (b) Torsional spring inserted in the cavity of a leg. (c) Spoke frame securing the gears and the springs. (d) Connection between the wheel and the motor shaft.

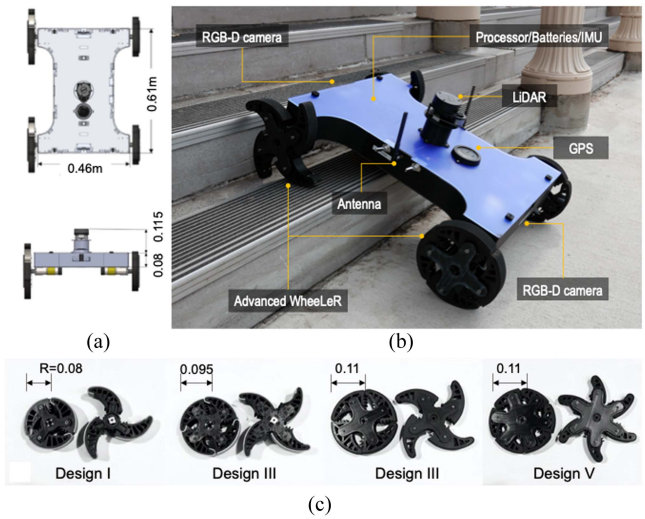


Fig. 15. Final α-WaLTR platform. (a) CAD model of the chassis in isometric, top, and front views. (b) Fully constructed hardware platform. (c) Four types of wheels.

1) Main processor: Jetson TX2 with a 64-bit quad-core Arm Cortex -A57 CPU and a 256-core NVIDIA Pascal GPU.
2) Actuators: Four of 5202 Series Yellow Jacket Planetary Gear Motors (188:1 Ratio, 30 RPM, 3.3–5-V Encoder) with torque of 3470 oz · in (24.5 N· m), each embedded with a quadrature encoder.
3) Sensors: Two RGB-Depth (RGB-D) cameras (Intel RealSense D-435i), inertial measurement unit (IMU) (Pixhawk Orange Cube; triaxial accelerometer, gyroscope, and compass), and GPS (CubePilot Here3 Precision GNSS Module).
4) Battery: 16 000-mAh 14.8-V LiPo battery for the drive system and 5 200-mAh 14.8-V LiPo battery for the sensors and the main processing board.
5) Motor controller: Two Basicmicro's Roboclaw 30-A motor controllers.
6) Others: Powered USB hub, power brick mini, 5 and 12-V dc regulators, and RP-SMA cables.

The front camera, LiDAR, and IMU are connected to a single powered USB hub powered by a 5200-mAh battery with a 5-V dc regulator. The IMU is powered through the power brick mini. The USB hub is connected to the USB 3.0 port, where the mini USB port is used to operate the rear camera. The 16 000-mAh battery powers the drive system and Jetson TX2 through a 12-V dc regulator. Between the motor controller and the main processing board, one power line is for the encoder system and the other for the motors. Two motors on each side of the robot are controlled using a single signal line and, thus, operate simultaneously. The reverse polarity SMA (RP-SMA) cables are used as wireless network extension cables attached to the Jetson TX2 board. The 2.4/5-GHz dual-band RP-SMA antennas are attached on both the sides of the robot for enhanced WiFi connectivity.

B. Design, Fabrication, and Assembly

The advanced WheeLeR mechanism in Design III consists of a central driving gear, four leg segments, four torsional springs, and two spoke frames for the selected four-leg configuration. The estimated stiffness of the spring k is about 0.9 N · m/rad with estimated $\mathcal{M}_r = 13$. Following the selection process previously described and based on commercial availability, we selected torsional springs that can work for both the wheel sizes with a wire diameter of 0.00216, an outer diameter of 0.019, and a torque of 1.45 N · m. Individual wheel components were 3-D-printed with PLA with 40% infill rate. The contact surface of each leg is covered with friction enhancing rubber sheet [see Fig. 14(a)]. The sheet is attached to the 3-D-printed leg using screws and adhesive. A torsional spring is inserted in the cavity in each leg, and all the legs are arranged around the central gear [see Fig. 14(b)]. Two spoke frames hold the springs in place and assemble the central gear and the legs [see Fig. 14(c)]. The fully assembled wheels are, then, attached to the motor shaft through a barrel hub [see Fig. 14(d)]. This connector allows easy and quick replacement of the wheel when needed.

Fig. 15(a) shows the CAD model of the robot in top and front views. While maintaining the overall dimensions of the chassis suggested from the optimization process, the curved chassis design provides sufficient space for the front and the rear motor mounts and the installation of the antennas while reducing the overall weight. The chassis comprises a custom-cut

carbon fiber baseplate, 3-D-printed PLA walls, and a laser-cut acrylic top cover. Each wall slides in a slot on its adjacent wall for secure locking between the two. The front and rear walls have windows for the RGB-D cameras, and the top cover has 3-D-printed LiDAR and GPS mounts. Two columns mounted on the baseplate support the top cover. Two antennas are installed on the left and right sides of the robot. Four 3-D-printed twist locks secure the top cover to the chassis walls, offering easy access to the internals of the robot for debugging and modifications. Additional screws and nuts are used for fully securing the top cover before deployment. Fig. 15(b) shows fully constructed α-WaLTR. Fig. 15(c) shows four sets of wheels built for physical experiment.

VI. Experimental Evaluation

Physical experiments focused on evaluating 1) the effect of the advanced WheeLeR design in vibration reduction; 2) versatile locomotion performance; and 3) autonomous stair climbing capability.

A. Evaluation of the Advanced WheeLeR Design in Vibration Reduction

To evaluate the effect of the torsional springs on reducing overall vibrations experienced by the robot, we employed the robot, equipped with the wheels in the advanced WheeLeR design and those in the previous design, and measured the acceleration responses, which are often considered to evaluate the effect of a suspension system on reducing vibrations in mobile robots [72], [73]. Pixhawk's embedded algorithm was adopted for measuring vibrations [61]. This algorithm first collects raw x, y, and z acceleration values from the IMU and processes them using a high-pass filter at 5 Hz to remove the robot's movement. This process establishes the baseline for each vibration axis. Differences between the measured and baseline data are captured, and the standard deviation is calculated for each axis value representing the corresponding vibration.

The experiments involved the robot equipped with the Design III wheels in two different sizes ($R = 0.095$ and 0.11), with and without torsional springs. By removing the torsional springs from their encasing (see Fig. 14), the mechanism becomes the same as the previous design [9]. The torsional springs are added to reduce vibrations, while the robot moves in the legged configuration. The experiments were performed on a rough surface, where the legs can remain open while moving. For a reference, we also measured the vibrations when the robot moves with regular (nontransformable) wheels in the same sizes. The robot equipped with each set of wheels was operated at three different speeds, 1.6, 3.2, and 4.5 rad/s. The vibration data were recorded for 30 s from Pixhawk. The average value for each axis was calculated, resulting in three values, a_x, a_y, and a_z. The whole-body vibration (WBV) was then calculated by WBV$= \sqrt{(a_x^2 + a_y^2 + a_z^2)}$ (ISO 2631-1:1997) as a measure of the overall vibrations experienced by the robot.

Fig. 16 shows the results in six groups of three bars in blue, orange, and yellow. Blue bars indicate the WBV values when the robot moves with regular wheels; orange bars are from the robot equipped with the advanced WheeLeR mechanisms; Yellow bars display the results from the robot equipped with the previous WheeLeR mechanisms, which do not have embedded torsional springs. The WBV measured from the robot with the advanced WheeLeR design was 25% ($R = 0.095$) and 33% ($R = 0.11$) less than that with the previous WheeLeR design when operated at the speed of 1.6 rad/s, 13% and 18% at 3.2 rad/s, and 7% and 9% at 4.5 rad/s. The effect of the torsional springs in vibration reduction was significant when the robot moved at a low speed, allowing sufficient time for the springs to absorb the shock while walking. When operating at a higher speed, the legs slip through without reaching the full contraction of the springs, resulting in less significant vibration reduction.

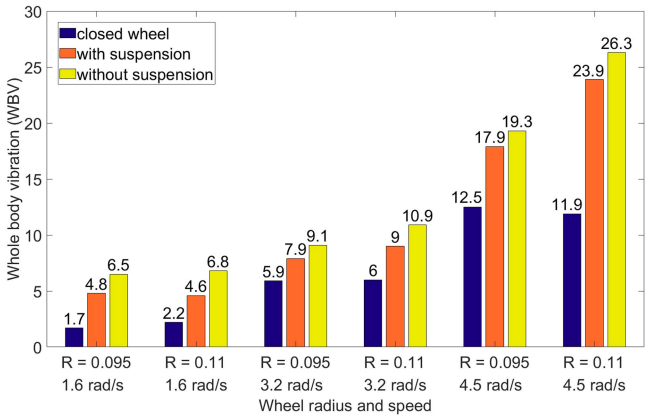


Fig. 16. Vibration analysis of α-WaLTR equipped with Design III ($R = 0.095$ and 0.11) with and without spring suspension.

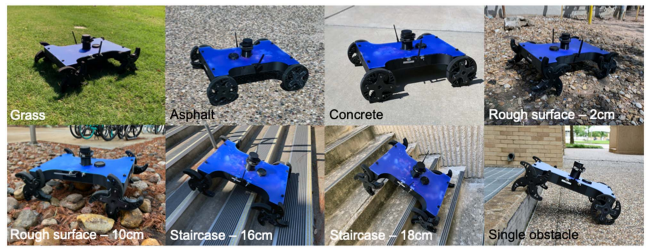


Fig. 17. Selected locomotion test environments: α-WaLTR was tested on grass, asphalt, concrete, rough surfaces with the average roughness height of 0.02, 0.05, and 0.1, staircases with the raise height of 0.16 and 0.18, and single obstacles with varying heights (0.08–0.24).

B. Evaluation of Versatile Locomotion Performance

For the physical evaluation of locomotion performance, α-WaLTR was remotely controlled to move on various terrains and climb over obstacles. Testing environments included grass, asphalt, concrete, rough terrain with overall roughness of 0.02 and 0.05, staircases (raise height/tread width: $0.16/0.42$; $0.18/0.29$), and single right-angled obstacles ($H_o = 0.16$–0.24) (see Fig. 17). On each environment, α-WaLTR moved forward at 3.2 rad/s three times, and successful traversals were counted.

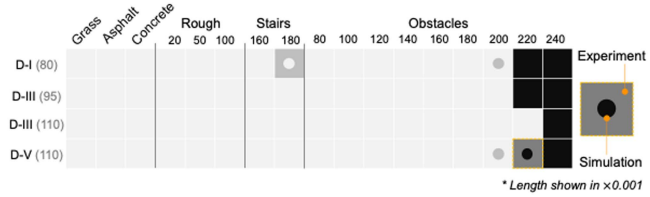


Fig. 18. Locomotion test results from physical experiments (and simulations shown in circles for comparison) for design I ($R = 0.08$), Design III (0.095 and 0.11), and Design V (0.11). 3/3 success traversals shown in tinted white, 2/3 in light gray, 1/3 in dark gray, and 0/3 in black.

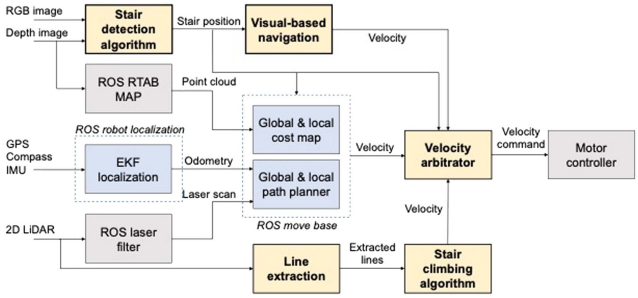


Fig. 19. ROS-based software architecture for autonomous navigation and stair climbing with newly developed algorithms shown in yellow boxes.

On grass, asphalt, concrete, and rough surfaces, the robot was tested for forward and rotational motions.

Fig. 18 visualizes the results from both physical experiments and simulations. Each square cell shows the experimental result for a specific wheel selection and test environment. A colored dot on a cell indicates the simulation outcome if it differs from the experimental result. The same color-coding scheme used in Fig. 12 was adopted. α-WaLTR showed a 3/3 success rate on grass, asphalt, concrete, and all rough terrains with $h = 20$, 50, and 100. On rough terrains, stairs, and obstacles, the wheels transformed into the legged configuration. The robot could reliably climb over both types of staircases except when it was equipped with Design I, showing a 2/3 success rate on the staircase with the raise height of 0.18. Design I ($R = 0.08$), Design III ($R = 0.095$), and Design V ($R = 0.11$) could reliably climb over an obstacle up to $H_o^{\max} = 0.2$, and Design III ($R = 0.11$) was able to climb up to 0.22. As shown previously in Fig. 3(b), when n_{leg} increases from Design I to V, $\mathcal{S}_c$ slightly decreases while C increases: $\mathcal{S}_c = 2.5$ and $C = 4$ for Design I, $\mathcal{S}_c = 2.1$ and $C = 5$ for Design III, and $\mathcal{S}_c = 1.92$ and $C = 7$ for Design V. This indicates that given R, a design with a smaller n_{leg} achieves better climbability due to the longer stroke expected from the legs when opened. When operating in the wheeled mode, $O^{\max}$ was about the same as the radius of the wheel resulting in $\mathcal{S}_c \approx 1$. The experimental results were highly consistent with the simulation outcomes, demonstrating the simulator's potential for predicting the physical locomotion performance of different wheel designs.

We also tested the robot's locomotion performance in turning on the spot for the Design III wheels. On grass, asphalt, concrete, and rough surfaces, both sets of Design III wheels showed similar performance measured by $\mathcal{S}_t$, such that $\mathcal{S}_t = 0.42/0.44$ on concrete, $0.38/0.44$ on asphalt, $0.32/0.36$ on a rough surface with $h = 0.02$, and $0.31/0.29$ with $h = 0.05$. Compared to the simulation results in Section IV-D, physical turning experiments resulted in $\mathcal{S}_t$ values 33% lower than that from the simulations. Owing to the passive nature of the transformable wheels, turning on the spot often results in two wheels on one side moving in one direction in the wheeled mode and the other two moving the opposite direction in the legged mode. In simulations, the motors perform ideally and continuously rotate even while walking. However, in reality, when the legs hit the ground, the angular speed of the motors instantly decreases drastically and recovers over time, increasing T_t and thus lowering $\mathcal{S}_t$ in physical experiments. However, overall trends of $\mathcal{S}_t$ in both simulations and experiments were consistent, implying that the simulations can provide useful performance indicators for different terrain conditions.

VII. Algorithms for Autonomous Navigation and Stair Climbing

Urban environments involve unique locomotion challenges due to coexisting man-made and natural environments with diverse obstacles. Staircases, meaning connectivity and accessibility for humans, often become a major barrier for wheeled robots. If a robot can traverse staircases, it is likely to climb over various obstacles and traverse diverse terrain types, as shown in our simulations and physical experiments. Therefore, the ability to traverse stairs is considered a signature capability of α-WaLTR and used to demonstrate the system-level integration. This section describes the overall software architecture with employed ROS packages and embedded algorithms to enable α-WaLTR to autonomously detect, navigate toward, and traverse a staircase. We adopted Design III ($R = 0.095$) in the stair detection and traversing experiments.

A. ROS-Based System Architecture

Fig. 19 shows the overall control software architecture based on the ROS. The control system combines the newly developed algorithms with open-sourced ROS packages to realize autonomous navigation functions, including localization, obstacle avoidance, path planning, and motor control. Localization utilizes the ROS Robot Pose Extended Kalman Filter (EKF) package based on the GPS, compass, and IMU data. Obstacle detection uses the laser and depth data captured by the 2-D LiDAR and RGB-D cameras. The Real-Time Appearance Based Mapping (RTAB-Map) package in the ROS calculates the visual odometry [60], and RTAB-Map generates point clouds of detected obstacles from the depth images captured by RGB-D cameras. Global and local cost maps are created using depth and laser data. The "move base" ROS package serves as the primary path planner [63]. This package processes the current velocity and position of the robot from the localization algorithm and generates several sample paths.

These adopted ROS packages are designed for conventional wheeled robots. One of the advantages of α-WaLTR is the

applicability of such extensive open-source control algorithms available for wheeled robots. However, to utilize the advanced locomotion capabilities of α-WaLTR, new algorithms must be added and modifications to existing ones are necessary. The newly developed algorithms include stair detection [64], visual-based stair navigation, stair climbing, and velocity arbitrator algorithms. The yellow boxes in bold text in Fig. 19 indicate these customized algorithms. The existing ROS move base package for the local cost map includes three classes: free space (cost: 0), obstacle space (cost: 254), and surrounding area of obstacle (cost: 253). The surrounding area is based on a user-defined distance from the detected obstacle and can be determined considering the required clearance and possibly the robot's operating speed.

A conventional wheeled robot can typically traverse low obstacles with the height up to the wheel radius R, while other factors also affect the actual climbing ability [1], [2]. The free space class in this case can be defined by the area with the obstacle height $H_o \leq R$ where $R = 0.2$. Upper and lower bounds of the classes can be adjusted based on the robot's maximum climbable obstacle height ($H_o^{\max}$). One may also consider a more detailed classification, e.g., dividing the free space into two classes—one for wheels and the other for legs. Each of these classes may have a uniquely assigned cost taking account for parameters like the power consumption and trajectory length.

B. Stair Detection Algorithm

The modified local cost map can be used for generating a path accounting the robot's locomotion capability. However, traversing a staircase requires the robot to utilize specialized algorithms, which are different from navigation algorithms for flat surfaces. During the stair-traversing process, the robot has to align itself with the stairs and maintain its heading orientation while climbing up or moving down. Therefore, the capability of autonomous stair detection and precise steering toward the entrance of the staircase is essential. This subsection presents an efficient stair detection strategy that can be applied for embedded robotic systems. While many machine-learning-based algorithms exist [65], [66], [67], [68], [69], [70], most of them are not directly suitable for embedded robotic applications due to 1) models trained using a significantly different dataset, resulting in low performance when implemented in the robot [64]; and 2) high computing and processing load for real-time operations and retraining when needed [64], [70].

We have recently developed a stair detection algorithm, which allows fast and reliable stair detection using RGB-D data [64]. In field operations, the environment can be significantly different from the data used in establishing the classification model. In this case, a model trained using an existing dataset requires retraining with a significant amount of new data. This is a costly and time-consuming process. The adopted stair detection method initially uses a support vector machine (SVM) to establish initial boundary conditions using an existing dataset. If the trained model does not satisfy the desired detection performance, these boundary conditions can be modified by a relatively small amount. This method is computationally efficient and fast, and it is, therefore, well suited for embedded mobile robotic applications. This stair detection algorithm is briefly reviewed here.

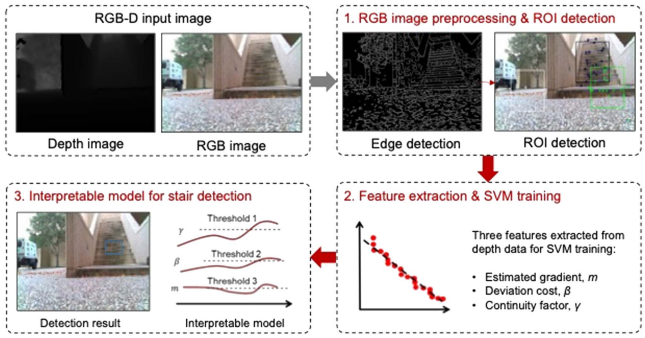


Fig. 20. Stair detection procedure using an interpretable classification model; reillustrated from [64].

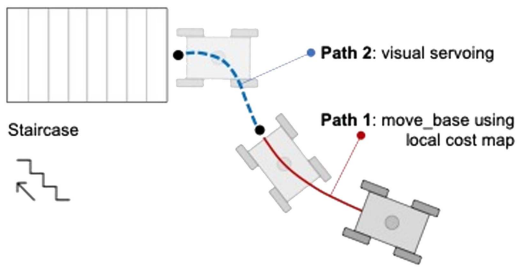


Fig. 21. Strategy for autonomous navigation toward the detected staircase using two sequential steps: following a path generated by the move base package (Path 1) and switching to visual servoing for real-time vision-based control (Path 2) for precise steering.

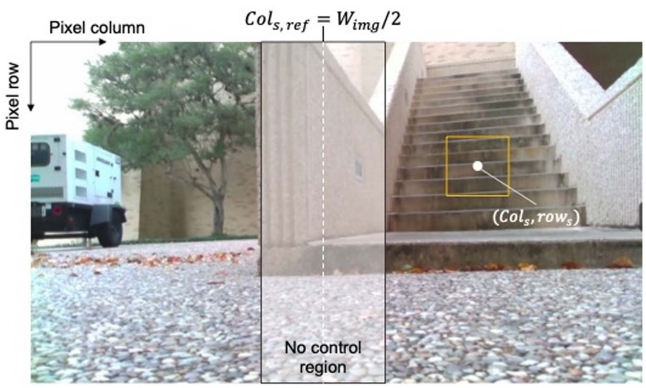


Fig. 22. Control scheme for visual servoing for steering the robot toward the staircase within close proximity.

Fig. 20 illustrates the three steps for stair detection, given RGB-D data: 1) defining regions of interest (ROIs) using the preprocessed RGB image; 2) extracting three features [i.e., estimated gradient (m), continuity factor (γ), and deviation cost (β)] from the depth data collected from the vertical middle line within individual ROIs; and 3) establishing an interpretable model for stair detection. The three features representing the characteristic of a staircase are inputted to the interpretable classification model to determine whether the image has a staircase or not. The evaluation performed in comparisons with an SVM-based

method [68] and You Look Only Once [70] show that the adopted method is comparable or better in stair detection performance while requiring less computational time and cost [64].

C. Stair Navigation Algorithm

Stair detection returns the position of the detected staircase, given by $\vec{p}_s = (x_s, y_s, z_s)^T$, which is the Cartesian coordinate of the center of the ROI with the detected staircase with respect to the reference frame fixed at robot's starting location. The pixel coordinate, $(\text{col}_s, \text{row}_s)$, is the column and row positions of the ROI relative to the upper left corner of the image as shown in Fig. 22. Upon detection of a staircase, the ROS move base package generates a path using $\vec{p}_s$ as the goal point to initialize the process. However, the localization accuracy can be affected by errors in the navigation system. For example, odometry calculations in robots with passively transformable wheels, such as α-WaLTR, are prone to errors. Another issue is the noise in the sensor data. In our case, inaccuracies in the depth measurements can result in an incorrect goal location. The depth data become more accurate as the robot gets closer to the stairs. This means that the robot can continuously update $\vec{p}_s$ in real time for generating updated paths. However, such a strategy requires significant processing time, causing delay. To address this, we implemented a sequential process of autonomous navigation toward the stairs. First, the robot follows the move base control command using the initial goal point $\vec{p}_s$ until the depth data returns the distance between the robot and the staircase is about 3 m (Path 1 in Fig. 21). Second, the robot switches its control strategy to visual servoing, which utilizes the pixel coordinates $(\text{col}_s, \text{row}_s)$ for steering the robot toward the staircase (Path 2 in Fig. 21).

The linear and angular velocity commands for visual servoing (u_x, u_{yaw}) are given by

$$u_x = \bar{u}_x; \quad u_{\text{yaw}} = -K_{\text{yaw}}(\text{col}_{s,\text{ref}} - \text{col}_s) \tag{12}$$

where $\bar{u}_x$ is a constant linear velocity command, K_{yaw} is a control gain for u_{yaw}, $\text{col}_{s,\text{ref}}$ is the reference column position, and col_s is the current column position of the stair on the image plane. In (12), the robot's yaw angle is controlled to match $\text{col}_{s,\text{ref}}$ to col_s. Fig. 22 shows the image from the robot's view. We used $\text{col}_{s,\text{ref}} = \frac{W_{\text{img}}}{2}$, where W_{img} is the image width. In this view, the ROI on the staircase is located on the right from $\text{col}_{s,\text{ref}}$, and thus, the robot is controlled to rotate in the clockwise direction. "No control region" is defined to avoid sensitive control. The range of "no control region" is from $\frac{W_{\text{img}}}{2} - 50$ to $\frac{W_{\text{img}}}{2} + 50$. If col_s is within this region, $u_{\text{yaw}} = 0$. Once the robot reaches the entrance of the staircase, it triggers the stair climbing algorithm, which is described below.

D. Autonomous Stair Climbing Algorithm

The presented algorithm enables autonomous stair climbing via wall tracing when there exists a wall(s) on one or both sides. It can be used for various staircases, including straight, L-shape, or U-shape staircases. The robot uses the laser data to trace the position and orientation of itself relative to the wall(s). It then controls the motors to keep itself aligned with the walls, maintaining a certain distance. When the robot reaches the landing floor, the robot may return to the ground floor using the same algorithm. The pseudo algorithm for this is provided in Algorithm 1, with the following notations: 1) d is the width of staircase; 2) State 0: ready to start next stair segment; 3) State 1: on a stair segment with double walls; 4) State 2: on a stair segment with single wall; 5) State 3: rotating toward next stair segment; and 6) State 4: finished stair climbing. Waiting for further commands. This algorithm simulated in Gazebo is shown in Fig. 23. A simulation environment was created based on the actual staircase we used for physical demonstration shown in Fig. 24 [see Fig. 23(a)]. The white dots in Fig. 23(b) represent the raw data collected by LiDAR, and the red lines indicate the extracted line segments [62]. Fig. 23(c)–(f) shows the climbing process and corresponding sensor data at each step.

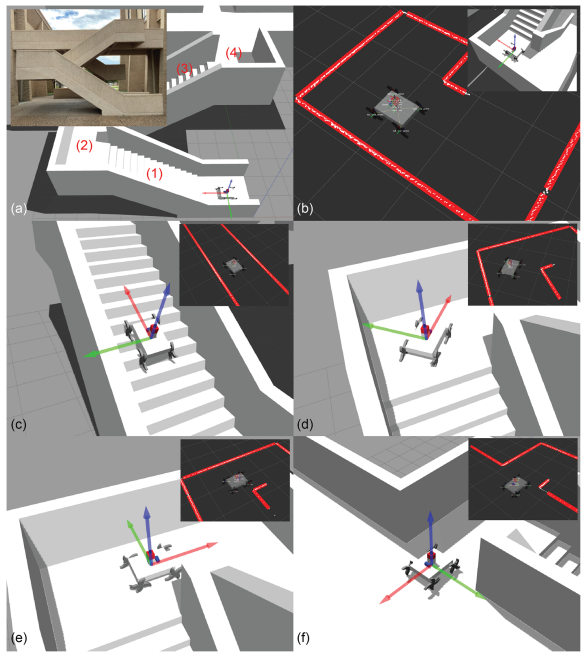


Fig. 23. (a)–(f) Gazebo simulation developed for U-shaped staircases replicating a physical environment and situations that α-WaLTR needs to tackle when climbing up the staircases using the presented algorithm.

This algorithm was developed for applications with expected environmental conditions. For stairs without walls, a different method may be adopted. For example, the stair climbing algorithm in [71] estimates the robot's heading angle from the information of stair edges obtained by RGB image processing and IMU. The robot controls its heading angle using this estimation while climbing a staircase. However, the performance of image processing may be affected by lighting conditions and other external factors (e.g., blurry images taken onboard while moving). When the staircase has a wall, the presented algorithm is expected to be more computationally efficient and robust, as it solely relies on laser data.

E. Physical Demonstration

We used α-WaLTR equipped with the Design III ($R = 0.095$) for system-level integration and physical demonstration of the

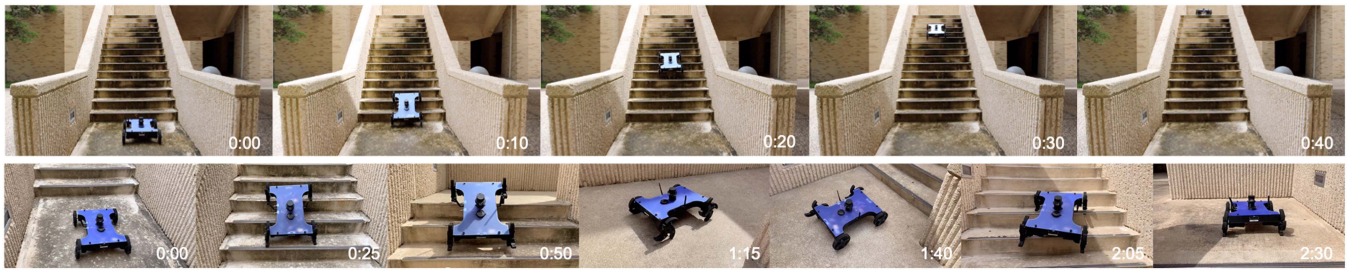
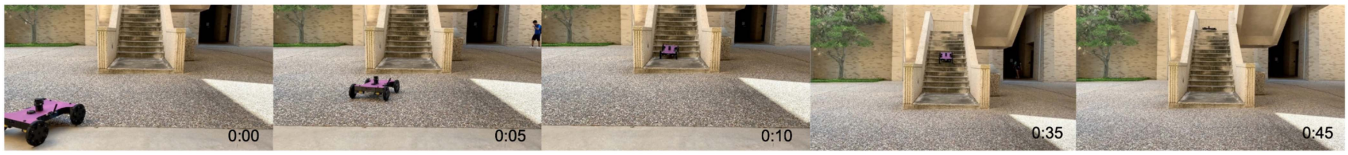


Fig. 24. Time-lapse sequence images from two separate trials of stair climbing on a U-shaped double staircase [see Fig. 23(a)] using the embedded algorithm presented in the Appendix.

(a)



(b)

Fig. 25. Autonomous stair detection, navigation, and climbing. (a) Once the robot is near a staircase, it uses visual servoing to move toward the detected staircase. (b) Integrated demonstration of stair navigation and climbing in time lapse sequence.

embedded autonomous functions. The selected environment was a U-shaped double staircase with a 0.18-m raise and 0.29-m tread width, consisting of two sets of double-walled stairs connected with a U-shaped landing floor. For onboard real-time autonomous navigation, a simple algorithm that utilizes LiDAR for real-time navigation on staircases was developed and implemented. The robot can be initially positioned in any direction. If the robot approaches stairs facing backward, it first turns around to utilize the legged mode while climbing; otherwise, it proceeds to the staircase. The 2-D LiDAR scans the walls and controls the drive system to align the robot in between the two walls while moving forward by autonomously adjusting its heading direction. When the robot reaches the landing floor, it determines the turning direction by examining the surrounding walls and navigates through the corners to find the next staircase.

The successful traversal rate using this algorithm was over 90% out of 20 trials. Too large of an offset at the heading angle resulted in the robot failing to adjust its orientation while traversing. We also experienced occasional hardware issues such as disconnected cables. Fig. 24 shows the time-lapse sequence images from two successful runs. Fig. 25 shows physical demonstration of the entire sequence of autonomous stair detection, navigation, and climbing. On the far left in Fig. 25(a), the robot's view shows the detected staircase, which governs the local navigation. Fig. 25(b) shows the time lapse sequence of the visual-based navigation followed by autonomous stair climbing.

VIII. Conclusion and Discussion

In this article, we presented an optimal design methodology, locomotion simulator, hardware development, physical experiments, and demonstration of a novel mobile robotic platform, α-WaLTR. The robot's versatile locomotion capabilities were enabled by passively transformable wheels that can transition from wheels to legs (and vice versa) based on the driving direction and environmental conditions. α-WaLTR is one of the largest mobile platforms equipped with passively transformable wheels. The multiobjective optimization analysis presented in this article allows the designer to select upper level design variables considering application-specific constraints. The developed Unity-based simulator enables comprehensive and comparative analyses among varying design options, especially when 1) the searching scopes for individual variables are large and 2) conventional optimization techniques are not applicable. The simulation results were closely aligned with the experimental outcomes, showing the potential of this simulator for predicting physical locomotion performance. This simulator can significantly reduce development time and cost for such robots. The system-level integration was demonstrated by the robot autonomously climbing over a staircase using a simple wall-tracing algorithm.

The spring–suspension mechanism newly introduced to the wheel design resulted in a meaningful reduction in overall vibrations. The torsional springs encased in individual leg segments

Algorithm 1: Stair Traversing Algorithm.

Reach the entrance of staircase and face toward the first stair segment
$state = 0$
for Every time step **do**
  if $state == 0$ **then**
    if Detects double-side walls **then**
      Start tracking both walls
      $state = 1$
    else if Detects only one wall on left/right side **then**
      Start tracking this wall
      $state = 2$
    else
      $state = -1$
    end if
  end if
  while $state == 1$ or $state == 2$ **do**
    Go forward/upward, while maintaining reasonable relative orientation and position w.r.t. the two walls [see Fig. 23(c)]/single wall [see Fig. 23(e)] being tracked
    ifDetects a wall in front within $d/2$ **then**
      Start tracking this wall, determine rotation direction toward next stair segment.
      $state = 3$
    else if Surpassed the walls/wall being tracked, cannot detect wall in front within d **then**
      $state = 4$
    end if
  end while
  while $state == 3$ **do**
    Rotate toward next stair segment [see Fig. 23(d)]
    if The wall being tracked is parallel to the orientation of robot **then**
      $state = 0$ [see Fig. 23(f)]
    end if
  end while
  while $state == 4$ **do**
    Stair climbing accomplished.
  end while
end for

kept the design simple and modular. This feature becomes more useful in the design with a small n_{leg}, where the entire body would suffer from more significant shocks while walking than the one with a small n_{leg}. With a larger n_{leg}, the walking behavior is much smoother, and the original WheeLeR mechanism without the springs can be used. A conventional spring–damper mechanism commonly adopted in cars and mobile robots may replace this spring-only mechanism for more effective shock absorption and improved long-term durability, especially for a larger platform. However, this design would increase the overall structural complexity. The rubber sheets attached to the wheels to increase friction may be replaced with properly designed tires for better shock absorption. Custom tires for individual legs may be designed and fabricated via 3-D printing or molding and casting.

The current platform has a maximum speed of 0.3 m/s. This speed may be acceptable for many applications but not for high-speed operations. α-WaLTR requires relatively high-torque motors compared to a conventional wheeled robot counterpart. The developed robots are intended to operate as part of a large swarm system, where individual robots are relatively small and inexpensive while satisfying locomotion objectives to operate in urban environments (e.g., traversing rough terrains and stairs). The presented wheel mechanism is simple and modular to accommodate easy maintenance, repair, and replacement when needed. We used low-cost off-the-shelf motors with high torque and low speed in the presented platform. Increasing the speed limit without sacrificing the torque typically increases the size and weight of the motor, creating a tradeoff to consider. For example, a lightweight robot may be equipped with high-speed low-torque motors with reduced payload for agile locomotion.

WheeLeR is designed to roll in the wheel configuration on flat smooth surfaces and transform into the legged configuration when traversing rough terrains or overcoming obstacles. However, its passive nature does not allow the accurate prediction of the transformation. Our current control strategy relies on the assumption that the robot moves with the closed wheels on flat surfaces and, thus, utilizes an extensive body of existing research outcomes on wheeled robot control. However, the robot cannot avoid occasional wheel-leg transformation in some wheels, especially when it makes a sharp turn. In this case, it deviates from the planned trajectory (which assumed wheeled locomotion). On rough surfaces or when traversing stairs or other obstacles, path planning based on the wheeled robot kinematics cannot accurately predict the actual trajectory of the robot. To control the robot's moving trajectory precisely, we must consider the differential drive kinematic model of the robot equipped with WheeLeRs and methods for detecting the configuration (open versus closed) of individual wheels. The kinematic model may account for the flexibility of the tire materials and kinematic constraints to develop tailored motion control strategies, as described in [74]. In this work, the violation of kinematics is modeled as a lump parameter [74]. To apply this strategy, an observer will need to estimate the lump parameter in real time. As we expect the robot to move in the legged locomotion on rough irregular terrains, the actual motions of individual wheels and the robot are hard to model. The kinematic model of the terrain itself may need to be developed together with the wheel and robot models [75]. The current control system is based on the move base package in the ROS combined with a set of custom-developed algorithms to compensate for the deviation from the planned path at each waypoint and to continuously update its local pathway. The embedded local obstacle avoidance algorithm allows the robot to detour around the nontraversable obstacles while following the trajectory.

REFERENCES

[1] G. Thomas and V. Vantsevich, "Wheel-terrain-obstacle interaction in vehicle mobility analysis," *Veh. Syst. Dyn.*, vol. 48, pp. 139–156, 2010.

[2] R. Siegwart, P. Lamon, T. Estier, M. Lauria, and R. Piguet, "Innovative design for wheeled locomotion in rough terrain," *Robot. Auton. Syst.*, vol. 40, nos. 2/3, pp. 151–162, 2002.

[3] F. Grimminger et al., "An open torque-controlled modular robot architecture for legged locomotion research," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3650–3657, Apr. 2020.

[4] E. Celaya and J. M. Porta, "A control structure for the locomotion of a legged robot on difficult terrain," *IEEE Robot. Autom. Mag.*, vol. 5, no. 2, pp. 43–51, Jun. 1998.

[5] R. Yajima, K. Nagatani, and Y. Hirata, "Research on traversability of tracked vehicle on slope with unfixed obstacles: Derivation of climbing-over, tipping-over, and sliding-down conditions," *Adv. Robot.*, vol. 33, no. 20, pp. 1060–1071, 2019.

[6] S. Al-Milli, L. D. Seneviratne, and K. Althoefer, "Track–terrain modelling and traversability prediction for tracked vehicles on soft terrain," *J. Terramech.*, vol. 47, no. 3, pp. 151–160, 2010.

[7] J. S. Koh and K. J. Cho, "Omega-shaped inchworm-inspired crawling robot with large-index-and-pitch (LIP) SMA spring actuators," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 2, pp. 419–429, Apr. 2013.

[8] Z. Liu et al., "Somatosensitive film soft crawling robots driven by artificial muscle for load carrying and multi-terrain locomotion," *Mater. Horiz.*, vol. 8, pp. 1783–1794, 2021.

[9] C. Zheng and K. Lee, "WheeLeR: Wheel-leg reconfigurable mechanism with passive gears for mobile robot applications," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 9292–9298.

[10] D. Lu, E. Dong, C. Liu, M. Xu, and J. Yang, "Design and development of a leg-wheel hybrid robot "HyTRo-I"," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 6031–6036 .

[11] E. Smith, E. Anwar, and I. Farkhatdinov, "Design and validation of mobile robot with hybrid wheel-leg design," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*2021, pp. 4682–4687.

[12] M. Lacagnina, G. Muscato, and R. Sinatra, "Kinematics, dynamics and control of a hybrid robot Wheeleg," *Robot. Auton. Syst.*, vol. 45, nos. 3/4, pp. 161–180, 2003.

[13] M. C. Birch et al., "A miniature hybrid robot propelled by legs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Expanding Soc. Role Robot. Next Millennium*, 2001, vol. 2, pp. 845–851.

[14] S. Hong, M. Wu, J. Xiao, X. Xu, and H. Lu, "Kylin: A transformable track-wheel hybrid robot," in *Proc. IEEE Int. Conf. Adv. Mechatronic Syst.*, 2017, pp. 7–12.

[15] J. Kim, Y. G. Kim, J. H. Kwak, D. H. Hong, and J. An, "Wheel & track hybrid robot platform for optimal navigation in an urban environment," in *Proc. SICE Annu. Conf.*, 2010, pp. 881–884.

[16] S. Mizunuma, K. Motomura, and S. Hirose, "Development of the arm-wheel hybrid robot "Souki-II"(Total system design and basic components)," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 4535–4540 .

[17] N. G. Bourbakis, "Kydonas—An autonomous hybrid robot: Walking and climbing," *IEEE Robot. Autom. Mag.*, vol. 5, no. 2, pp. 52–59, Jun. 1998.

[18] J. Zhao, T. Han, S. Wang, C. Liu, J. Fang, and S. Liu, "Design and research of all-terrain wheel-legged robot," *Sensors*, vol. 21, no. 16, 2021, Art. no. 5367.

[19] L. Bruzzone, M. Baggetta, S. E. Nodehi, P. Bilancia, and P. Fanghella, "Functional design of a hybrid leg-wheel-track ground mobile robot," *Machines*, vol. 9, no. 1, 2021, Art. no. 10.

[20] Z. Chen, J. Li, J. Wang, S. Wang, J. Zhao, and J. Li, "Towards hybrid gait obstacle avoidance for a six wheel-legged robot with payload transportation," *J. Intell. Robot. Syst.*, vol. 102, no. 3, pp. 1–21, 2021.

[21] T. Okada, W. T. Botelho, and T. Shimizu, "Motion analysis with experimental verification of the hybrid robot PEOPLER-II for reversible switch between walk and roll on demand," *Int. J. Robot. Res.*, vol. 29, no. 9, pp. 1199–1221, 2010.

[22] J. M. Morrey, B. Lambrecht, A. D. Horchler, R. E. Ritzmann, and R. D. Quinn, "Highly mobile and robust small quadruped robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, vol. 1, pp. 82–87.

[23] P. Arena, S. De Fiore, L. Patané, M. Pollino, and C. Ventura, "Insect inspired unsupervised learning for tactic and phobic behavior enhancement in a hybrid robot," in *Proc. Int. Joint Conf. Neural Netw.*, 2010, pp. 1–8.

[24] K. Nagatani, M. Kuze, and K. Yoshida, "Development of transformable mobile robot with mechanism of variable wheel diameter," *J. Robot. Mechatronics*, vol. 19, pp. 252–253, 2007.

[25] Y. She, C. J. Hurd, and H. J. Su, "A transformable wheel robot with a passive leg," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 4165–4170.

[26] T. Sun, X. Xiang, W. Su, H. Wu, and Y. Song, "A transformable wheel-legged mobile robot: Design, analysis and experiment," *Robot. Auton. Syst.*, vol. 98, pp. 30–41, 2017.

[27] A. Mardani, S. Ebrahimi, and K. Alipour, "6AP wheel: A new transformable robotic wheel for traction force improvement and halting avoidance of a UGV on soft terrains," *Mech. Des. Struct. Mach.*, vol. 50, pp. 3370–3385, 2022.

[28] D. Y. Lee, S. R. Kim, J. S. Kim, J. J. Park, and K. J. Cho, "Origami wheel transformer: A variable-diameter wheel drive robot using an origami structure," *Soft Robot.*, vol. 4, no. 2, pp. 163–180, 2017.

[29] D. Y. Lee, J. K. Kim, C. Y. Sohn, J. M. Heo, and K. J. Cho, "High-load capacity origami transformable wheel," *Sci. Robot.*, vol. 6, no. 53, 2021, Art. no. eabe0201.

[30] N. Moriya, H. Shigemune, and H. Sawada, "A robotic wheel locally transforming the diameter for the locomotion on rough terrain," in *Proc. IEEE Int. Conf. Mechatronics Autom.*, 2020, pp. 1257–1262.

[31] A. J. Clark, K. A. Cissell, and J. M. Moore, "Evolving controllers for a transformable wheel mobile robot," *Complexity*, vol. 2018, pp. 1–12, 2018.

[32] S. C. Chen, K. J. Huang, W. H. Chen, S. Y. Shen, C. H. Li, and P. C. Lin, "Quattroped: A leg–wheel transformable robot," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 2, pp. 730–742, Apr. 2014.

[33] M. Ning et al., "Design, analysis, and experiment for rescue robot with wheel-legged structure," *Math. Probl. Eng.*, vol. 217, 2017, Art. no. 5719381.

[34] W. H. Chen, H. S. Lin, Y. M. Lin, and P. C. Lin, "TurboQuad: A novel leg–wheel transformable robot with smooth and fast behavioral transitions," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1025–1040, Oct. 2017.

[35] K. P. Sun, D. C. Yang, X. Chang, P. X. Lu, Z. H. U. Heng, and C. H. E. N. Ken, "Design and development of the wheel-legged robot dog kara," *DEStech Trans. Comput. Sci. Eng.*, pp. 164–169, 2019.

[36] X. Zhao, W. Su, and S. Zhang, "The simulation and analysis of a new rescue robot with transformable wheels," in *Proc. 2nd World Conf. Mech. Eng. Intell. Manuf.*, 2019, pp. 545–548.

[37] K. Kim, Y. Kim, J. Kim, H. S. Kim, and T. Seo, "Optimal trajectory planning for 2-DOF adaptive transformable wheel," *IEEE Access*, vol. 8, pp. 14452–14459, 2020.

[38] Y. Kim, Y. Lee, S. Lee, J. Kim, H. S. Kim, and T. Seo, "STEP: A. new mobile platform with 2-DOF transformable wheels for service robots," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 4, pp. 1859–1868, Aug. 2020.

[39] H. Y. Chen, T. H. Wang, K. C. Ho, C. Y. Ko, P. C. Lin, and P. C. Lin, "Development of a novel leg-wheel module with fast transformation and leaping capability," *Mech. Mach. Theory*, vol. 163, 2021, Art. no. 104348.

[40] Y. Chang, Z. Zhang, and C. Wang, "Control system design based on CANopen network for a transformable multimodal robot," in *Proc. IEEE Int. Conf. Mechatronics Autom.*, 2018, pp. 948–953.

[41] F. Zhou, X. Xu, H. Xu, T. A. Zou, and L. Zhang, "Transition mechanism design of a hybrid wheel-track-leg based on foldable rims," *Proc. Inst. Mech. Eng., C, J. Mech. Eng. Sci.*, vol. 233, no. 13, 2019, pp. 4788–4801.

[42] Z. Luo, J. Shang, G. Wei, and L. Ren, "A reconfigurable hybrid wheel-track mobile robot based on Watt II six-bar linkage," *Mech. Mach. Theory*, vol. 128, pp. 16–32, 2018.

[43] Z. Wei, G. Song, Y. Zhang, H. Sun, and G. Qiao, "Transleg: A wire-driven leg-wheel robot with a compliant spine," in *Proc. IEEE Int. Conf. Inf. Autom.*, 2016, pp. 7–12.

[44] A. Torisaka, K. Eguchi, S. Miura, V. Parque, and T. Miyashita, "Leg-circle transformable wheel for improved runnability of a lunar rover," *J. Intell. Mater. Syst. Struct.*, vol. 32, no. 9, pp. 1013–1023, 2021.

[45] K. Tadakuma et al., "Mechanical design of the wheel-leg hybrid mobile robot to realize a large wheel diameter," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 3358–3365.

[46] İ. Mertyüz, A. K. Tanyıldızı, B. TaşarA. B. Tatar, and O. Yakut, "FUHAR: A transformable wheel-legged hybrid mobile robot," *Robot. Auton. Syst.*, vol. 133, 2020, Art. no. 103627.

[47] Q. Xu, W. Guo, M. Sherikar, C. Wu, and L. Wang, "Environment perception and motion strategy for transformable legged wheel robot on rough terrains," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2018, pp. 2153–2158.

[48] Y. S. Kim, G. P. Jung, H. Kim, K. J. Cho, and C. N. Chu, "Wheel transformer: A miniaturized terrain adaptive robot with passively transformed wheels," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 5625–5630.

[49] Y. S. Kim, G. P. Jung, H. Kim, K. J. Cho, and C. N. Chu, "Wheel transformer: A wheel-leg hybrid robot with passive transformable wheels," *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 1487–1498, Dec. 2014.

[50] S. Ryu, Y. Lee, and T. Seo, "Shape-morphing wheel design and analysis for step climbing in high speed locomotion," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1977–1982, Apr. 2020.
[51] L. Bai, J. Guan, X. Chen, J. Hou, and W. Duan, "An optional passive/active transformable wheel-legged mobility concept for search and rescue robots," *Robot. Auton. Syst.*, vol. 107, pp. 145–155, 2018.
[52] S. Zhang, J. T. Yao, Y. B. Wang, Z. S. Liu, Y. D. Xu, and Y. S. Zhao, "Design and motion analysis of reconfigurable wheel-legged mobile robot," *Defence Technol.*, vol. 18, no. 6, pp. 1023–1040, 2021.
[53] 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.
[54] V. B. Bhandari, *Design of Machine Elements*. New York, NY, USA: McGraw-Hill, 1994.
[55] P. Y. Papalambros and D. J. Wilde, *Principles of Optimal Design*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[56] W. A. Mattingly, D. J. Chang, R. Paris, N. Smith, J. Blevins, and Ouyang, "Robot design using unity for computer games and robotic simulations," in *Proc. 17th Int. Conf. Comput. Games*, 2012, pp. 56–59.
[57] W. Meng, Y. Hu, J. Lin, F. Lin, and R. Teo, "ROS unity: An efficient high-fidelity 3D multi-UAV navigation and control simulator in GPS-denied environments," in *Proc. 41st Annu. Conf. IEEE Ind. Electron. Soc.*, 2015, pp. 002562–002567.
[58] A. Hussein, F. Garcá, and C. Olaverri-Monreal, "ROS and unity based framework for intelligent vehicles control and simulation," in *Proc. IEEE Int. Conf. Veh. Electron. Saf.*, 2018, pp. 1–6.
[59] J. J. Roldán et al., "Multi-robot systems, virtual reality and ROS: Developing a new generation of operator interfaces," in *Robot Operating System.*New York, NY, USA: Springer, 2019, pp. 29–64.
[60] M. Labbé and F. Michaud, "Long-term online multi-session graph-based SPLAM with memory management," *Auton. Robots*, vol. 42, no. 6, pp. 1133–1150, 2018.
[61] Ardupilot, 2021. [Online]. Available: https://ardupilot.org/copter/docs/common-measuring-vibration.html
[62] Laser Line Extraction, 2021. [Online]. Available: https://github.com/kam3k/laser_line_extract-ion
[63] Move Base 2016. [Online]. Available: https://github.com/ros-planning/navigation
[64] K. Lee, V. Kalyanram, C. Zheng, S. Sane, and K. Lee, "Vision-based ascending staircase detection with interpretable classification model for stair climbing robots," in *Proc. Int. Conf. Robot. Autom.*, 2022, pp. 6564–6570.
[65] S. Wang and H. Wang, "2D staircase detection using real AdaBoost," *Proc. 7th Int. Conf. Inf. Commun. Signal Process.*, 2009, pp. 1–5.
[66] S. Wang, H. Pan, C. Zhang, and Y. Tian, "RGB-D image-based detection of stairs, pedestrian crosswalks and traffic signs," *J. Vis. Commun. Image Represent*, vol. 25, no. 2, pp. 263–272, 2014.
[67] X. Qian and C. Ye, "3D object recognition by geometric context and Gaussian-mixture-model-based plane classification," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 3910–3915.
[68] R. Munoz, X. Rong, and Y. Tian, "Depth-aware indoor staircase detection and recognition for the visually impaired," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, 2016, pp. 1–6.
[69] S. Ponnada, S. Yarramalle, and M. R. Tv, "A hybrid approach for identification of manhole and staircase to assist visually challenged," *IEEE Access*, vol. 6, pp. 41013–41022, 2018.
[70] U. Patil et al., "Deep learning based stair detection and statistical image filtering for autonomous stair climbing," in *Proc. 3rd IEEE Int. Conf. Robot. Comput.*, 2019, pp. 159–166.
[71] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, D. M. Helmick, and L. Matthies, "Autonomous stair climbing for tracked vehicles," *Int. J. Robot. Res.*, vol. 26, no. 7, pp. 737–758, 2007.
[72] Z. Li, X. Jing, B. Sun, and J. Yu, "Autonomous navigation of a tracked mobile robot with novel passive bio-inspired suspension," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 6, pp. 2633–2644, Dec. 2020.
[73] J. Moreno, E. Clotet, M. Tresanchez, D. Martínez, J. Casanovas, and J. Palacín, "Measurement of vibrations in two tower-typed assistant personal robot implementations with and without a passive suspension system," *Sensors*, vol. 17, no. 5, 2017, Art. no. 1122.
[74] C. A. P. Fernández, J. J. F. Cerqueira, and A. M. N. Lima, "Nonlinear trajectory tracking controller for wheeled mobile robots by using a flexible auxiliary law based on slipping and skidding variations," *Robot. Auton. Syst.*, vol. 118, pp. 231–250, 2019.
[75] N. Chakraborty and A. Ghosal, "Kinematics of wheeled mobile robots on uneven terrain," *Mech. Mach. Theory*, vol. 39, no. 12, pp. 1273–1287, 2004.

Chuanqi Zheng received the B.S. degree in welding technology and engineering and the M.S. degree in materials engineering from the Harbin Institute of Technology, Harbin, China, in 2012 and 2014, respectively. He is currently working toward the Ph.D. degree in mechanical engineering with Texas A&M University, College Station, TX, USA.

His research interests include multiterrain mobile robots, swarm robotics, and distributed communication and control algorithms.

Siddharth Sane received the B.Tech. degree in mechanical engineering with Honors in machine design from the College of Engineering Pune, Pune, India, in 2019, and the M.S. degree in mechanical engineering from Texas A&M University, College Station, TX, USA, in 2021.

He is currently with Texas A&M University. His research interests include mobile robotics, material handling systems, mechanism design, and optimization.

Kangneoung Lee received the B.S. and M.S. degrees in mechanical engineering from Sungkyunkwan University, Seoul, South Korea, in 2015 and 2017, respectively. He is currently working toward the Ph.D. degree in mechanical engineering with Texas A&M University, College Station, TX, USA.

From 2017 to 2020, he was a Research Engineer with Mando Corporation, Seongnam, South Korea. His research interests include computer vision, path planning, and control of mobile robots.

Vishnu Kalyanram received the B.Eng. degree in production engineering from the PSG College of Technology, Coimbatore, India, in 2019. He is currently working toward the master's degree in mechanical engineering with Texas A&M University, College Station, TX, USA.

His research interests include applications of computer vision in robotic and mechatronic systems.

Kiju Lee (Member, IEEE) received the B.S.E. degree in electrical and electronics engineering from Chung-Ang University, Seoul, South Korea, in 2002, and the M.S.E. and Ph.D. degrees in mechanical engineering from Johns Hopkins University, Baltimore, MD, USA, in 2006 and 2008, respectively.

From 2008 to 2019, she was with the Faculty of the Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, OH, USA. Since 2019, she has been holding a joint faculty position between the Department of Engineering Technology and Industrial Distribution and the Department of Mechanical Engineering, Texas A&M University, College Station, TX, USA. She is the Director of the Adaptive Robotics and Technology Lab through which she runs various research projects focusing on swarm robotics, novel robotic mechanism design, and human–robot interaction.