Ringbot: Monocycle Robot With Legs

Kevin Genehyub Gim, *Member, IEEE*, and Joohyung Kim, *Member, IEEE*

***Abstract*—This article presents the development and evaluation of Ringbot, a novel leg-wheel transformer robot incorporating a monocycle mechanism with legs. Ringbot aims to provide versatile mobility by replacing the driver and driving components of a conventional monocycle vehicle with legs mounted on compact driving modules inside the wheel. The article covers the hardware and software implementation of a prototype robot. The Ringbot prototype features a wheel and two driving modules located inside, each equipped with a 3-DoF leg for balancing, steering, and legged motions to assist monocycle driving. The driving control is achieved through decoupled speed controller and steering controller. In addition, active-legged motions are implemented and managed through a finite-state machine. The controllers for wheeled driving and legged motions were tested in a simulation environment, as well as on the hardware prototype, to verify the concept of a monocycle with legs and evaluate the prototype's capabilities.**

***Index Terms*—Field robots, legged robots, mechanism design, wheeled robots, wheel-leg transformable robots.**

I. Introduction

THE escalating attention towards autonomous transportation has instigated research into novel mobile robot technologies possessing dynamic navigation skills for real-world applications from both academia and industry. The two predominant modes of movement can be categorized as wheeled and legged. The wheeled systems, usually preferred for their proficiency on flat surfaces, demonstrate an edge in efficiency, stability, and speed compared to legged systems. Yet, legged systems demonstrate their versatility by traversing diverse terrains and structures that challenge wheeled mechanisms. In addition, they offer improved maneuverability and exhibit the ability to adapt their gait to meet specific task requirements.

As a result of this tradeoff, there have been active efforts to investigate hybrid leg-wheel mechanisms, aiming to harness the benefits of both modes of locomotion. During the 2015 DARPA Robotics Challenge, multiple robots demonstrated smart strategies by transitioning between legged walking and wheeled driving [1], [2], [3]. By doing so, these robots were able to maneuver more efficiently on paved surfaces and traverse uneven terrain or stairs, resulting in a strong performance throughout the competition.

Manuscript received 26 July 2023; revised 16 December 2023; accepted 20 January 2024. Date of publication 6 February 2024; date of current version 28 February 2024. This paper was recommended for publication by Associate Editor K. Lee and Editor M. Yim upon evaluation of the reviewers' comments. *(Corresponding author: Joohyung Kim.)*

The authors are with the KIMLAB (Kinetic Intelligent Machine LAB), University of Illinois Urbana-Champaign, Urbana, IL 61801 USA (e-mail: kevingim89@gmail.com; robozzang@gmail.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TRO.2024.3362326, provided by the authors.

Digital Object Identifier 10.1109/TRO.2024.3362326

One of the most popular forms of leg-wheel hybrid robots is a wheeled foot. By having passive or active wheeled feet, the legged robot can glide and skate on flat terrain. Moritz et al. [4] proposed an optimization-based algorithm to address complex design and motion synthesis for passive wheeled-foot robots. Researchers at ETH Zurich developed an active wheeled-foot quadruped robot that can perform hybrid locomotion [5], [6]. This includes not only wheeled movement, but also four-legged walking while applying brakes on the wheels. The papers demonstrated the effectiveness of a wheeled-legged hybrid locomotion strategy, revealing that wheeled skating is significantly more efficient than quadruped walking, allowing faster maneuverability on flat terrain. In addition, the robot was able to traverse inclined terrains and steps using quadruped walking motion.

In addition, humanoid and biped robots have adopted the wheeled-foot concept to minimize the complexity and risks associated with legged movement while improving efficiency. Biped robots equipped with wheeled feet have been a focus of research [7], [8]. These designs primarily employed wheeled locomotion similar to a wheeled-inverted pendulum, with the legs modifying the position or height of each wheel. This functionality enhances steering capabilities and offers greater maneuverability compared to traditional bipedal walking.

Furthermore, some robots have integrated wheeled-foot quadruped legs with a humanoid torso, aiming for stable locomotion while leveraging the advantages of wheeled-legged hybrid movement. An example is the CENTAURO robot, developed at Istituto Italiano di Tecnologia (IIT), which features four 5-Degrees of Freedom (DoF) legs, each with a wheel at its end [9]. Its humanoid torso equips it for demanding manipulation tasks and offers significant physical resilience, all within a human-like size and weight. Kim et al. [10] have also contributed to this field by developing a wheel-leg mechanism used for a humanoid lower body, consisting of four 2-DoF legs and mecanum wheels as feet. Though the leg's DoF is reduced, the incorporation of the mecanum wheel mechanism enables omnidirectional movement and the ability to lift the legs, facilitating traversal over a variety of terrains.

Other studies have pursued switching between wheeled and legged locomotion by developing transformable robots. Several robots have been designed to convert their wheels into passive legs for navigating challenging terrain [11], [12]. These robots exhibit a comparable configuration and operational concept, possessing four wheels that can transform into noncircular-shaped legs. In the legged mode, the wheels continue to rotate as they would in wheeled mode, but the noncircular shape of the legs enables alternating contact with the ground. This feature allows the robot to tackle terrains challenging for regular

Fig. 1. Ringbot, a monocycle robot with legs.

wheels. The wheel-legged robot developed by Wang et al. [13] incorporates a complex linkage mechanism into its wheels. This mechanism transforms the wheels into 2-DoF legs, enabling a walking motion with alternating steps in each leg, rather than merely rotating the wheels.

Moreover, there have been developments in quadruped robots capable of transforming their legs into wheels or spheres, facilitating active quadruped walking in legged mode as well as rolling movement. One such biomimetic robot, inspired by a spider, can both crawl and roll using its legs [14]. This robot features four large legs that can reconfigure into a single large wheel encircling the body and push the ground to facilitate rolling. A similar concept has been applied to a spherical robot that transforms from a quadruped to a sphere for rolling [15]. This robot possesses four legs covered with a spherical shell, which retracts into the wheel for its spherical mode. In this mode, an internal driving unit shifts the weight inside the sphere, controlling its rolling movement.

In this work, we got motivation from how humans use their legs when operating compact wheeled vehicles in real life. The critical function of human legs becomes evident in managing vehicles with small footprints, like motorcycles or scooters. Not only do they actively establish equilibrium by grounding the foot, but they also prevent potential falls and navigate the vehicle beyond the wheel's limitations. This active leg involvement efficiently counteracts the inherent stability challenges found in such compact vehicles.

This research introduces a novel concept in wheeled-leg hybrid locomotion platforms: A robot that merges a monocycle robot with integrated legs, named the Ringbot. The primary focus of this project is to develop a Ringbot prototype, integrating both hardware and software implementation, to validate the new mechanism concept and demonstrate its effectiveness. The Ringbot prototype, shown in Fig. 1, consists of a ring-shaped wheel and two driving modules located inside, like two hamsters in a hamster wheel. Each driving module has a leg on top with 3-DoF, which can be used for balancing and steering during wheeled driving by shifting the center of mass (CoM) laterally, but also performing various legged motions, such as falling recovery, self-right, and in place turning. This combination offers unique and practical functionality, highlighting the potential of the proposed mechanism for real-world applications that require versatile locomotion capabilities across diverse terrains, including urban environments and indoor settings. One example of its potential application is in last-mile delivery services.

The rest of this article is organized as follows. Section II provides an overview of the monocycle mechanism and its characteristics. In Section III, the kinematic and dynamic modelings of the proposed robot system are presented. Section IV explains the hardware implementation in detail. Section V covers the software implementation, including the driving controllers and the legged motion generator. Simulation results are presented in Section VI, and hardware experiments are displayed in Section VII. Finally, Section VIII concludes this article.

II. Background—Monocycle

A monocycle is a one-wheeled vehicle similar to a unicycle, first conceptualized in the 19th century and further developed as a compact motorized transportation. It features a large wheel housing the driving components and a driver who controls the vehicle by adjusting the throttle and shifting weight laterally. The large-single wheel provides an outstanding ability to overcome obstacles while maintaining a minimal footprint.

Since the monocycle has only one wheel, it requires support or balancing in a lateral direction when the wheel is stationary or moving at a slow speed. However, once the wheel gains sufficient angular velocity, the gyroscopic effect comes into play, stabilizing the lateral dynamics and preventing it from falling over. On the other hand, in the longitudinal direction (pitch), the monocycle's structure makes it intrinsically stable.

The inherent pitch stability of the monocycle is due to its metacentric weight distribution, which ensures that the CoM is located lower than the wheel's rotational axis. This characteristic is similar to the stability observed in a floating ship, where the opposing forces of gravity and buoyancy metacenter prevent the vessel from rolling over.

In the context of bipedal walking, humans also inherently form a metacenter at the intersection point of the ground reaction force (GRF) [16]. Even though static biped standing is inherently unstable, resembling an inverted pendulum, the creation of a metacentric virtual pivot point through sufficient GRF transforms biped walking into a virtual pendulum, thereby enhancing stability against perturbations. By applying this concept, a planar running dynamics model called Trunk spring-loaded inverted pendulum was introduced and applied to the control of rimless wheel robot and planar biped running robot [17].

Fig. 2 presents models for comparing a unicycle and a monocycle. As the unicycle has a higher CoM position than the wheel's rotating axis around which the entire body rotates, it is inherently unstable in a wheeled inverted pendulum configuration, requiring actuation power and control effort to make the robot balanced all the time. In contrast, the monocycle's rotating center is positioned above the CoM, making the system intrinsically stable. Consequently, the monocycle requires

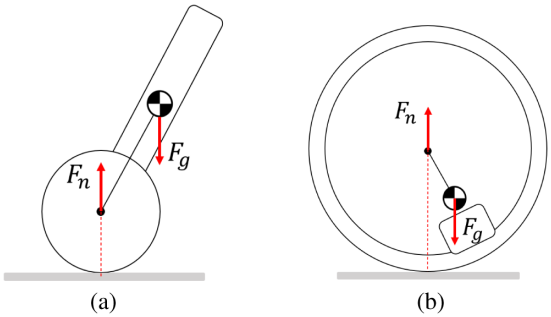


Fig. 2. Mass distribution and applied forces in single wheel mechanisms. (a) Unicycle model. (b) Monocycle model.

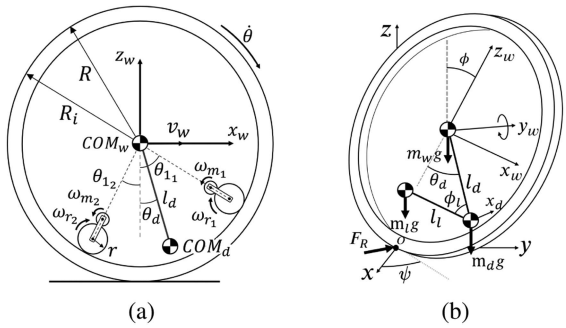


Fig. 3. Diagrams for 2-D and 3-D modeling of Ringbot. (a) Side view 2-D model. (b) Simplified 3-D model.

less speed and energy to maintain stability over the unicycle configuration [18]. In addition, the large wheel size offers several advantages, such as better obstacle-overcoming capability compared to smaller wheels and ample space inside of the wheel for potentially incorporating other mechanisms without compromising stability.

There have been several monocycle robots developed in the past. Some studies have explored the use of a flywheel to create a gyroscopic effect that stabilizes a monocycle and tilts the flywheel to cause a precession effect for steering [19], [20], [21]. Another study has investigated using a ballast as a pendulum [22]. By shifting the ballast mechanism laterally, the robot can maintain balance under small external disturbances.

Previous studies have explored a variety of design and control strategies for mobile robots using the monocycle mechanism. However, there are certain limitations to this approach that make it challenging to apply in practical situations. While the gyroscopically stabilized monocycles exhibited good balance performance, the system was found to be energy-inefficient, as it constantly required the generation of large momentum to the flywheel, even when the robots were stationary. The use of the internal mechanism as a ballast for balancing was also found to have limitations, as it could only handle very small angle deviations for balancing. In addition, the internal space of the wheel in the previous robots was occupied by the driving and balancing mechanisms, leaving no room for payloads or other mechanisms that could potentially bring practical functionalities to the system.

To address these limitations, we propose a new monocycle robot that utilizes compact driving modules equipped with legs.

III. Wheeled Driving Modeling

This section derives the mathematical modeling for Ringbot. The kinematic constraints between the wheel and driving modules are presented, and the nonholonomic rolling condition is defined. In addition, equations of motion of simplified 2-D and 3-D models are also derived in the following subsections.

A. Driving Kinematics

Fig. 3(a) shows a diagram describing the side view of Ringbot. Note that the 2-D diagram does not include the legs but just the wheel and driving modules, as it only accounts for motions in the sagittal plane of the wheel. The two driving modules travel independently inside the wheel with the speed of the driving motor, ω_{m_1} and ω_{m_2}, respectively. When the system is in a steady state that $\dot{\theta}$ is constant and $\omega_{m_1} = \omega_{m_2}$, a relationship between the wheel's rolling speed and the driving motor speed can be expressed as follows:

$$\begin{aligned}\dot{\theta} = \omega_{r_1} &= -N_w N_d \omega_{m_1} \\ &= \omega_{r_2} = -N_w N_d \omega_{m_2}\end{aligned} \tag{1}$$

where ω_{r_i} is the driving gear speed of ith driving module. The driving gear speed ω_{r_i} can be calculated from the motor speed ω_{m_i} and the gear ratio, where N_d represents the gear ratio of the transmission within the driving module, and N_w represents the gear ratio between the wheel and the driving module, calculated as $N_w = \frac{r}{R_i}$, the ratio between the wheel's inner radius and the driving module gear pitch radius.

If the system is not in a steady state, the wheel and driving module have the following kinematic relationship:

$$\begin{aligned}\dot{\theta} &= \omega_{r_1} - \dot{\theta}_{1_1} \\ &= \omega_{r_2} - \dot{\theta}_{1_2}.\end{aligned} \tag{2}$$

In Fig. 3(a), CoM_d denotes the position of the combined CoM of the two driving modules. The movement of CoM_d can be considered as a point-mass connected on a linkage l_d that rotates with respect to the y_w axis. The corresponding joint angle θ_d can be defined from the pitch angles of each driving module θ_{1_1} and θ_{1_2}

$$\theta_d = \frac{\theta_{1_1} + \theta_{1_2}}{2}. \tag{3}$$

Besides, as the monocycle is a nonholonomic system, the rolling constraint can be defined assuming an ideal rolling at a contact point without slipping

$$\begin{bmatrix}\dot{x} \\ \dot{y}\end{bmatrix} = R\dot{\theta}\begin{bmatrix}\cos\psi \\ \sin\psi\end{bmatrix} \tag{4}$$

where ϕ is the yaw angle of the wheel.

B. Driving Dynamics

The equations of motion of Ringbot were derived using the simplified 3-D model as shown in Fig. 3(b). The simplified 3-D dynamics model includes a wheel and two-point mass attached at mass-less linkages. Each point mass represents the CoM of the two driving modules and the CoM of the legs, respectively. At the center of the wheel, a revolute joint connects the driving module CoM with the linkage l_d that rotates along the y_w axis. The leg CoM is attached at the linkage l_l, connected at the driving module CoM with a revolute joint rotates along the x_d axis.

Using the simplified dynamic model, the design parameters of the prototype can be determined by considering the dynamics of the system. The equations of motion of the 3-D model were derived through the Newton–Euler method.

In Fig. 3(b), the world coordinate system is $[x, y, z]$. The yaw and roll angles of the wheel are represented with ψ and ϕ, respectively. Frame $[x_w, y_w, z_w]$ is a moving frame attached at the wheel's center. Note that the wheel's frame only reflects the yaw and roll angle; keep the pitch angle zero.

By applying ψ and ϕ with a 3-1 Euler angle sequence, the angular velocity of the moving wheel frame can be represented as follows:

$$\begin{aligned}\omega_{x_w} &= \dot{\phi}\\ \omega_{y_w} &= \dot{\psi} s\phi\\ \omega_{z_w} &= \dot{\psi} c\phi. \end{aligned} \tag{5}$$

Note that s and c are the abbreviations of sin and cos, respectively. The abbreviations are used in the rest of the article. Adding the rolling angular velocity $\dot{\theta}$, the angular velocity of the rotating wheel, Ω can be defined as

$$\begin{aligned}\Omega_{x_w} &= \omega_{x_w} = \dot{\phi}\\ \Omega_{y_w} &= \omega_{y_w} + \dot{\theta} = \dot{\psi} s\phi + \dot{\theta}\\ \Omega_{z_w} &= \omega_{z_w} = \dot{\psi} c\phi. \end{aligned} \tag{6}$$

As the rate of change of linear momentum is equal to the forces and the rate of change of angular momentum is equal to the torques

$$\begin{aligned} F &= \dot{p}_s = \dot{p}_w + \omega \times p_w\\ T &= \dot{L}_s = \dot{L}_w + \omega \times L_w \end{aligned}$$

where p_s is the linear momentum about the fixed space and p_w is the linear momentum of the moving reference frame. Also, L_w is the angular momentum of the moving frame, calculated as $L_i = I_i \Omega_i$. The force and force relation can be written as

$$\begin{aligned} F_{R_{x_w}} &= \dot{p}_{x_w} + \omega_{z_w} p_{y_w} - \omega_{y_w} p_{z_w}\\ F_{R_{y_w}} - Mgs\phi &= \dot{p}_{y_w} + \omega_{x_w} p_{z_w} - \omega_{z_w} p_{x_w}\\ F_{R_{z_w}} - Mgc\phi &= \dot{p}_{z_w} + \omega_{y_w} p_{x_w} - \omega_{x_w} p_{y_w}\\ F_{R_{y_w}} R + \tau_{l_x} + \tau_{d_x} + \tau_{R_l} c\theta_d &= I_{x_w} \dot{\Omega}_{x_w} + I_{y_w} \Omega_{y_w} \omega_{z_w} - I_{z_w} \Omega_{z_w} \omega_{y_w}\\ - F_{R_{x_w}} R + \tau_{l_y} + \tau_{d_y} + \tau_{R_d} &= I_{y_w} \dot{\Omega}_{y_w} + I_{z_w} \Omega_{z_w} \omega_{x_w} - I_{x_w} \Omega_{x_w} \omega_{z_w}\\ \tau_{l_z} + \tau_{d_z} + \tau_{R_l} s\theta_d &= I_{z_w} \dot{\Omega}_{z_w} + I_{x_w} \Omega_{x_w} \omega_{y_w} - I_{y_w} \Omega_{y_w} \omega_{x_w} \end{aligned} \tag{7}$$

where $F_{R_{x_w}}, F_{R_{y_w}}, F_{R_{z_w}}$ are the reaction forces acting on the contact point O, and M is the total mass of the robot, $M = m_w + m_d + m_l$. τ_{R_d} and τ_{R_l} are the reaction torque of the joint torque exerted on the driving module CoM and leg CoM joints, respectively. Also, τ_d and τ_l are the torque generated by the driving module CoM and leg CoM defined as follows:

$$\begin{aligned} \tau_{d_x} &= l_d m_d g c\theta_d s\phi\\ \tau_{d_y} &= -l_d m_d g c\phi s\theta_d\\ \tau_{d_z} &= -l_d m_d g s\phi s\theta_d\\ \tau_{l_x} &= m_l g s\phi (l_d c\theta_d - l_l c\phi_l c\theta_d) + l_l m_1\, g c\phi s\phi_l\\ \tau_{l_y} &= -m_l g c\phi s\theta_d (l_d - l_l c\phi_l)\\ \tau_{l_z} &= -m_l g s\phi s\theta_d (l_d - l_l c\phi_l). \end{aligned} \tag{8}$$

The legs are modeled as an inverted pendulum with a massless rod and a point mass at its end, representing the center of mass of the legs CoM_l, which is connected to the center of mass of the two driving modules CoM_d. It is designed to rotate only around the y_w axis, with the angle of rotation denoted by ϕ_l.

As the wheel is rolling on the ground, the linear momentum of the moving frame has relations with the angular momentum with the following constraints derived from (4):

$$\begin{aligned} p_{x_w} - MR\Omega_{y_w} &= 0\\ P_{y_w} + MR\Omega_{x_w} &= 0\\ P_{z_w} &= 0. \end{aligned} \tag{9}$$

Combining (5) to (9), the equations of motion expressed in the absolute orientation of the wheel $[\phi, \theta, \psi]^T$. can be obtained.

$$\begin{aligned} &(I_{x_w} + MR^2)\ddot{\phi} + (I_{y_w} + MR^2)(\dot{\psi} s\phi + \dot{\theta})\dot{\psi} c\phi\\ &\quad - I_{z_w} \dot{\psi}^2 c\phi s\phi = MgRs\phi + \tau_{l_x} + \tau_{d_x} + \tau_{R_l} c\theta_d \end{aligned} \tag{10}$$

$$\begin{aligned} &(I_{y_w} + MR^2)\left(\ddot{\psi} s\phi + \dot{\psi}\dot{\phi} c\phi + \ddot{\theta}\right) - MR^2 \dot{\psi}\dot{\phi} c\phi\\ &\quad = \tau_{l_y} + \tau_{d_y} + \tau_{R_d} \end{aligned} \tag{11}$$

$$I_{z_w} \ddot{\psi} c\phi - I_{y_w} \dot{\phi}\left(\dot{\psi} s\phi + \dot{\theta}\right) = \tau_{l_z} + \tau_{d_z} + \tau_{R_l} s\theta_d. \tag{12}$$

From the equations of motion, the critical velocity can be derived, which is the minimum velocity of the wheel such that the gyroscopic forces stabilize the wheel from a minor disturbance. Assume the wheel is rolling upright with a constant speed ($\phi_l = \theta_d = \tau_R = 0$, $\dot{\theta} =$ constant) with a little disturbance at the side, causing a small tilt angle $\phi = \alpha$.

By assuming $\dot{\psi}$ and $\dot{\alpha}$ are very small, higher order terms can be neglected. Then, (10) can be expanded to

$$\begin{aligned} &(I_{x_w} + MR^2)\ddot{\alpha} + (I_{y_w} + MR^2)\dot{\theta}\dot{\psi} c\alpha\\ &\quad = (MR + m_l(l_d - l_l) + m_d l_d) g\alpha \end{aligned} \tag{13}$$

$$I_{z_w} \ddot{\psi} c\alpha - I_{y_w} \dot{\alpha}\dot{\theta} = 0. \tag{14}$$

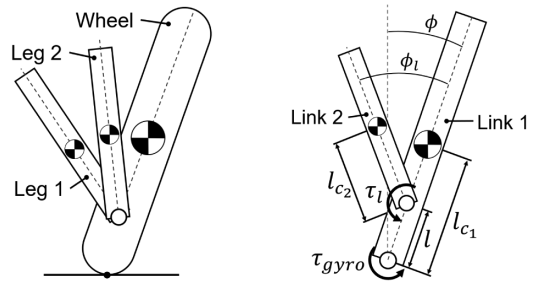


Fig. 4. Diagrams for lateral dynamics modeling of ringbot. (a) Frontal view of Ringbot. (b) Simplified 2-D model.

Integrating (14), we get

$$\dot{\psi} = \frac{I_{y_w}}{I_{z_w} c\alpha}\dot{\theta}\alpha. \tag{15}$$

Substitute $\dot{\psi}$ in (13) to get the following simplified equation:

$$(I_{x_w} + MR^2)\ddot{\alpha} = -\left[(I_{y_w} + MR^2)\frac{I_{y_w}}{I_{z_w}}\dot{\theta}^2 - (MR + m_l(l_d - l_l) + m_d l_d)\right]\alpha. \tag{16}$$

To ensure that (16) yields a stable solution, indicating that the roll angle will converge to zero and the wheel will maintain balance, the value within the square brackets must be positive.

$$\dot{\theta} > \sqrt{\frac{I_{z_w}(MR + m_l(l_d - l_l) + m_d l_d)}{I_{y_w}(I_{y_w} + MR^2)g}}. \tag{17}$$

If we consider the wheel as a thin hoop, $I_{x_z} = \frac{1}{2}MR^2$ and $I_{y_w} = MR^2$. By substituting the inertia terms, we can obtain the critical angular velocity as follows:

$$\dot{\theta} > \sqrt{\frac{(MR + m_l(l_d - l_l) + m_d l_d)g}{4MR^2}}. \tag{18}$$

C. Simplified Lateral Dynamics

The roll angle control is important for steering and balancing Ringbot, which can be achieved by shifting the legs laterally. As shown in Fig. 4, the lateral dynamic model of Ringbot can be simplified to a 2-DoF planar serial linkage by modeling the contact between the wheel and the ground as a 1-DoF revolute joint. In this model, the wheel and its two driving modules are treated as a single body, denoted as Link 1, while the two legs are combined into Link 2. It is important to note that leg 1 and leg 2 are symmetrical, each maintaining a constant angular offset from the midplane while in driving mode, except in cases where they reach the limits of their range of motion or make contact with the environment. When the robot is static, there is no torque acting on the first joint as it is just a ground contact point. In this case, the simplified model is equivalent to an Acrobot model, an underactuated 2-DoF inverted pendulum with an actuator at the second revolute joint [23]. By applying the Acrobot model, the state variables of the simplified model are defined as

$$X = \begin{bmatrix}\phi \\ \phi_l\end{bmatrix}. \tag{19}$$

Again, ϕ is the roll angle corresponding to the first joint angle which represents the ground contact point, and ϕ_l denotes the leg CoM roll joint angle as shown in Figs. 3(b) and 4(b).

Then, the manipulator equation is expressed as

$$\begin{bmatrix}M_{11} & M_{12} \\ M_{21} & M_{22}\end{bmatrix}\ddot{X} + \begin{bmatrix}C_{11} & C_{12} \\ C_{21} & 0\end{bmatrix}\dot{X} = \begin{bmatrix}\tau_{g_1} \\ \tau_{g_2}\end{bmatrix} + Bu \tag{20}$$

where M, C, τ_g, and B are

$$\begin{aligned}
M_{11} &= m_2 l_1^2 + 2m_2 l_1 l_{c_2} c\phi_l + m_2 l_{c_2}^2 + I_1 + I_2 \\
M_{12} &= m_2 l_{c_2}^2 + l_1 m_2 l_{c_2} c\phi_l + I_2 \\
M_{22} &= m_2 l_{c_2}^2 + I_2 \\
C_{11} &= -2l_1 l_{c_2} m_2 \dot{\phi}_l s\phi_l \\
C_{12} &= -l_1 l_{c_2} m_2 \dot{\phi}_l s\phi_l \\
C_{21} &= l_1 l_{c_2} m_2 \dot{\phi} s\phi_l \\
\tau_{g_1} &= m_2\, g(l_1 s\phi + l_{c_2} s(\phi + \phi_l)) + m_1\, g l_{c_1} s\phi \\
\tau_{g_2} &= m_2\, g l_{c_2} s(\phi + \phi_l) \\
B &= \begin{bmatrix}0 \\ 1\end{bmatrix}.
\end{aligned} \tag{21}$$

When the wheel is rolling with a nonzero angular velocity, the gyroscopic effect caused by the angular momentum acts in the opposite direction to gravity, preventing the wheel from tippling over. In (10), the gyroscopic effect is expressed as taking the term acting opposite to the gravity torque

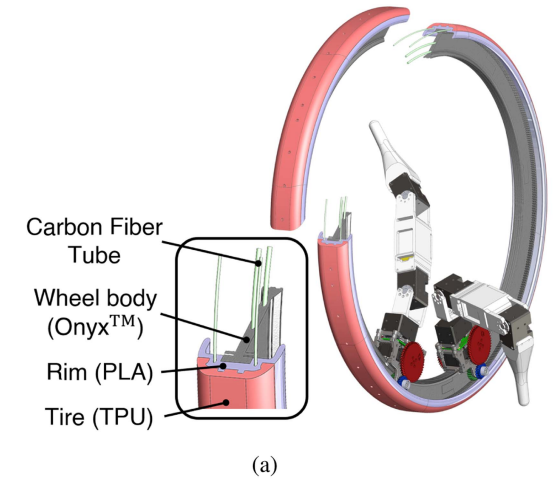
$$\tau_{gyro} = (I_{y_w} + MR^2)\dot{\theta}\dot{\psi}c\phi. \tag{22}$$

By manipulating the gravitational force through ϕ_l and the gyroscopic effect through angular velocity $\dot{\theta}$, the robot can control its roll angle beyond the upright equilibrium ($\phi = \phi_l = 0$).

IV. Hardware Implementation

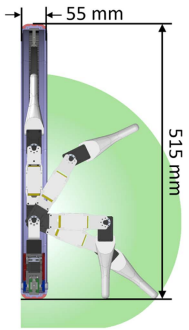
The structure of Ringbot is inspired by a monocycle vehicle composed of a large wheel and components for driving inside the wheel, including an engine, a driver's seat, and a driver. The driver manipulates the throttle to adjust speed and shifts weight laterally to steer and balance the vehicle.

Similarly, Ringbot consists of three main components: 1) a wheel; 2) driving modules; and 3) legs attached to each driving module. A 3-D CAD model for the overall hardware prototype is presented in Fig. 5(a). The most noticeable feature of Ringbot's appearance is the large ring-shaped wheel that wraps around all other parts. Inside the wheel, two driving modules travel along the wheel's inner surface. The two driving modules can be compared to two hamsters in a hamster wheel. On each driving module, one leg is attached. The legs are used to shift the CoM laterally for balancing and steering control. Furthermore,

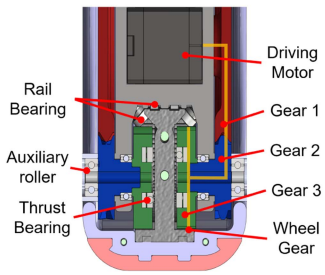


(a)

(b)



(c)



(d)

Fig. 5. Details of the Ringbot hardware. (a) Ringbot hardware prototype CAD model. (b) Close-up view of two driving modules. (c) Leg workspace in the frontal plane. (d) Cross-sectional view of a driving module and the gear transmission.

Ringbot is capable of performing legged motions using the legs. The specification of the implemented robot hardware is shown in Table I.

Details of the hardware design will be explained in the following subsections. To assess critical mobile robot specifications such as maximum range and operating time, Ringbot underwent a driving test on a 400-m running track. The focus was to determine the maximum distance and duration Ringbot could achieve in its wheeled mode on a single battery charge. Testing stopped once the voltage of any battery dropped below 3.4 V, indicating complete battery depletion. If the robot fell, it was manually reset to continue from the point of interruption.

TABLE I
SYSTEM SPECIFICATION

Wheel outer diameter	515 mm
Wheel inner diameter	500 mm
Total mass	2.532 kg
Wheel mass	1.145 kg
Driving module mass (w/o leg)	0.290 kg
Leg mass	0.404 kg
Driving module motor	XC330-M288-T (Customized)
Leg motor	XM430-W210-T
Operating Computer	Raspberry PI Zero 2W
IMU	BNO055 9-DoF IMU × 2
Battery	850mAh 11.1 V LiPo Battery × 2 300mAh 11.1 V LiPo Battery × 2
Maximum driving speed	5.0 km/h
Maximum driving range	3.0 km
Maximum driving duration	37 min

A. *Driving Module*

The two driving modules have identical configurations, except only one train has a single-board computer to operate the entire robot system. The main role of the driving modules is to shift the CoM in the longitudinal direction to accelerate/decelerate the wheel. Also, the driving modules become platforms for computers, sensors, and batteries needed to operate the robot system. Furthermore, the legs are attached to the top of the driving modules.

The selection of a motor to actuate the driving module is one of the most important design decisions. The primary design objective was to surpass a max speed of 4 km/h, faster than human walking speed. At the same time, a compact driving module size was pursued to keep the CoM lower as well as to provide more space for the leg linkages. An integrated servo motor package with a controller and sensors was preferred for concise electronics in limited space. To meet these demands, a compact-size servo motor was selected that has a built-in microcontroller that processes the internal feedback control as well as provides a serial network connection between all the connected motors.

As the single motor does not have sufficient power to generate enough torque and speed to reach the target speed, a single driving module employs two twin servo motors and synchronizes its actuation. By using two motors within one driving module, the driving torque can be enhanced. Consequently, the internal gearbox of the servo motors had custom modifications to increase the speed by 3.7 times compared to the original product in order to reach faster driving speed. The driving motors are powered at 7.5 V and operated in the pulse width modulation (PWM) control mode.

In addition, 3-D printed gears are attached to each servo motor output with the 2.61:1 gear ratio to amplify the speed of the motors. For efficient transmission, a herringbone gear design is applied to the 3-D printed gears.

Each driving module is equipped with a 9-DoF inertial measurement unit (IMU) that is capable of measuring absolute orientation, angular velocity, and linear accelerations. Combining the independent IMU measurements from each driving module makes it possible to estimate the current states of both the driving modules and wheels more accurately.

To facilitate tetherless, stand-alone operation, a single-board computer is installed on one of the driving modules and connected to all the motors and sensors in the Ringbot system. All the motors, including the leg motors, are connected through Transistor–Transistor Logic (TTL) serial communication. In addition, the IMUs are serially connected via I2C communication. Each driving module is powered by a 300 mAh three-cell Lithium Ion Polymer (LiPo) battery mounted at the driving module body and an 850 mAh three-cell LiPo battery at the leg linkage, respectively. A coil cable, which is elastic and capable of being stretched beyond the wheel's inner diameter, connects the two driving modules. This ensures that the distance between the two modules is not restricted by the cable. The motor communication lines and I2C lines are combined via the cable, enabling a single driving module's computer to manage the entire system.

As seen in Fig. 5(d), the driving module is mounted on the rail by six ball bearings that only allow for movement along the longitudinal direction of the rail. Two bearings roll along grooves at the top of the rail, while the other four bearings are angled at 45° and support the side of the rail to prevent any lateral movement. Gear 3 engages with the internal gear at the wheel and has thrust bearings inside that contact the wheel body's side wall to reduce slack between the wheel and driving modules. To further prevent deformation of the structural frame, auxiliary rollers are installed at the axis of Gear 2 to distribute stress all over the wheel structure. It is crucial to ensure a secure fit between the driving module and wheel body while minimizing friction, as any looseness or distortion can adversely impact driving performance and balance stability.

B. Wheel

The wheel is a crucial component of Ringbot, which allows the robot to move by rolling on the ground and houses all other elements. The wheel comprises three parts: 1) the wheel body; 2) rim; and the 3) tire. The wheel body is a structural frame for the wheel, providing a rail structure for the driving modules to travel on. In addition, it contains an internal gear that engages with the pinion gears of the driving modules. The rim serves as an intermediary piece between the wheel body and the tire. The tire was fabricated with an elastic material that directly contacts the ground. With a total width of 50 mm, the wheel has a 22-mm flat surface, providing more contact area for a greater stability margin.

The wheel diameter is one of the most important design parameters of the monocycle that determines driving performance and dynamics, as well as the space constraints for the mechanisms inside. To ensure a metacentric weight distribution and avoid collisions between the legs, we have established a minimum wheel size to accommodate four driving modules in the lower half of the wheel. As the wheel's inner diameter gets larger, it also necessitates a higher driving motor speed to achieve the angular speed that generates a gyroscopic effect to stabilize the wheel. Therefore, it is important to limit the wheel size to ensure the motors at the driving module can generate sufficient speed to exceed the critical velocity, which is the minimum angular velocity needed to stabilize a rolling wheel. The constraint for the maximum wheel radius can be determined by the equation for critical velocity derived from the equations of motion of the simplified dynamics model. For determining the wheel size upper limit, we neglect m_l and m_d and only consider the wheel's mass in (18). Combining (1) and (18), we can get the upper limit of the inner radius, which is equivalent to the pitch diameter of the ring gear inside of the wheel

$$R_i < \frac{4r^2 N_d^2 \omega_{m_{\max}}^2}{g}.$$

For the selected driving motor, the allowable range for the wheel diameter is between 230 mm and 1126 mm. After taking into account the safety factor and the fabrication feasibility, we have chosen an inner wheel diameter of 500 mm and the resulting outer diameter of 515 mm.

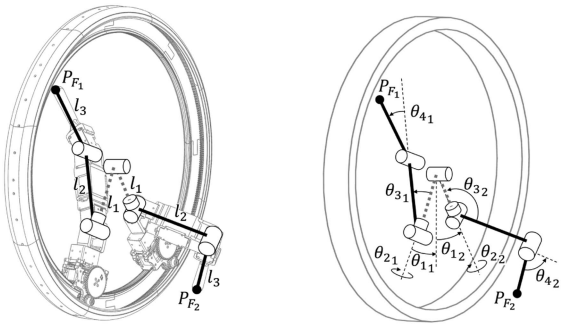
In pursuit of a high degree of freedom in part geometry and a quick manufacturing time, the wheel components are fabricated using 3-D printing technology. As the wheel is designed as a large ring-shape, it is crucial to make the wheel structure manage stress and impacts effectively despite the absence of spokes. As depicted in Fig. 5(a), the wheel body parts are printed using micro carbon fiber-filled nylon with continuous carbon fiber reinforcement layers in pursuit of enhanced stiffness and strength. The rim parts are printed with polylactic acid (PLA), and the tire parts are printed with thermoplastic polyurethane. Since the printing volume is limited, the wheel body, rim, and tire are divided into eight segments and assembled to form a single wheel. The assembly is constructed to prevent separation between parts by interlocking the joints and firmly connecting the tire and rim to the wheel body using screws. In addition, four carbon fiber tubes are inserted inside the wheel body and rim to reinforce the assembly and prevent separation due to impacts as well as reinforce the wheel structure.

C. Leg

Ringbot has two legs, each of which has 3-DoF and is mounted on top of each driving module. In Fig. 5(c), the workspace of a leg is marked with a green shaded area, taking into account the joint angle limit of each joint, but without considering internal collisions. The link length and joint configuration of each leg are designed to allow the tip of the leg is able to touch the ground while the wheel is standing upright, enabling the robot to perform various motions using the legs to assist the wheeled driving.

Three position-controlled servo motors actuate the leg, arranged in a Yaw-Roll-Roll configuration. The joint configuration of entire legs is presented in Fig. 6. The servo motors in the legs are controlled using the same serial communication as the driving module motors, which simplifies the electronic peripherals and the software framework while enabling coordinated control of both the leg and driving module.

In order for the robot to balance or steer using its legs as inverted pendulums, it is crucial that the legs have an effective



(a) (b)

Fig. 6. Joint configuration of Ringbot. (a) Leg joint diagram overlapped on the CAD drawing. (b) Name assignment for the leg joint angles.

TABLE II
DH PARAMETER TABLE FOR THE LEG

i-1	i	a	α	d	θ
0	1	0	$\frac{\pi}{2}$	0	θ_1
1	2	0	$\frac{\pi}{2}$	l_1	$\theta_2 - \frac{\pi}{2}$
2	3	0	$-\frac{\pi}{2}$	0	$\theta_3 + \frac{\pi}{2}$
3	4	l_2	0	0	θ_4
4	F	l_3	0	0	0

mass capable of influencing the wheel's dynamics. As the robot's structures are 3-D printed using PLA, it is challenging to achieve an effective mass without adding extra weights on the leg linkages. To address this, a LiPo battery is inserted into the first linkage of each leg, increasing the mass by 160% heavier compared to printing the same volume with solid infill. Moreover, having additional batteries greatly extends the robot's operational time compared to relying solely on the driving module batteries.

The presence of legs allows Ringbot to perform diverse tasks that distinguish it from conventional monocycle robots. In order to control the leg for various legged motions, the forward and inverse kinematics of the robot's legs are solved analytically. Fig. 6 illustrates how a driving module and its corresponding leg can be modeled as a 4-DoF robot arm mounted at the center of the wheel. The first joint represents the driving module's movement, which travels along the wheel's inner diameter and can be considered a revolute joint located at the wheel's center, with a linkage length of l_1. Table II displays the Denavit–Hartenberg (DH) parameter table for the leg, constructed using the modified DH parameter convention. Note that as the DH parameters of the two legs are identical, the subscript used to indicate the specific driving module has been omitted. This table enables the solution of forward kinematics, which calculates the foot's position based on joint angles.

An analytical solution for inverse kinematics has been obtained, enabling task-space control of the legs. The solution finds joint angles that satisfy a given foot position. Note that the first joint angle θ_{1_1} and θ_{1_2} are controlled by the driving controller. Therefore, these angles are not found by the inverse kinematics. The analytical solution finds the leg motor angles $[\theta_{2_i}, \theta_{3_i}, \theta_{4_i}]$ for the given foot position of ith leg, P_{F_i}.

The inverse kinematics solution for an arbitrary foot position $P_{F_i} = [P_{x_i}, P_{y_i}, P_{z_i}]$, can be obtained as follows:

$$\begin{aligned}\theta_{2_i} &= \text{atan2}(P_{x_i} c\theta_{1_i} + P_{z_i} s\theta_{1_i}, P_{y_i}) \\ \theta_{3_i} &= -\text{sign}(P_{y_i})\frac{\pi}{2} - (\gamma_i + \beta_i) \\ \theta_{4_i} &= -\pi + \text{acos}\left(\frac{l_2^2 + l_3^2 - a_i^2}{2 l_2 l_3}\right)\end{aligned} \tag{23}$$

where a_i, γ_i, and β_i stand for

$$\begin{aligned}a_i &= \sqrt{l_1^2 - 2 s\theta_{1_i} l_1 P_{x_i} + 2 c\theta_{1_i} l_1 P_{z_i} + P_{x_i}^2 + P_{y_i}^2 + P_{z_i}^2} \\ \beta_i &= \text{acos}\left(\frac{l_2^2 + a_i^2 - l_3^2}{2 l_2 a_i}\right) \\ \gamma_i &= -\text{atan}\left(\frac{l_1 + P_{z_i} c\theta_{1_i} - P_{x_i} s\theta_{1_i}}{P_{y_i} c\theta_{2_i} + P_{x_i} c\theta_{1_i} s\theta_{2_i} + P_{z_i} s\theta_{1_i} s\theta_{2_i} +}\right).\end{aligned} \tag{24}$$

Note that the inverse kinematic solution finds the elbow-up solution all the time.

V. SOFTWARE IMPLEMENTATION

The overall software implementation schematic for Ringbot can be found in Fig. 7. As depicted, Ringbot's software architecture is implemented using the robot operating system (ROS). Robot operating node is the main node that controls the Ringbot hardware. It includes a state estimator, controllers, and a finite state machine. The node communicates with the servo motors using TTL serial communication. Robot operating node runs at a loop frequency of 100 Hz. The IMU node manages I2C communication for the operation of IMUs, processes the collected data, and publishes them as a topic at a 100 Hz rate. To control the robot remotely, an Android smartphone app connects to the ROS master on Ringbot's system via WiFi. A pilot can command wheel speed and steering angle using a joystick user interface.

For the driving of the monocycle, the control states for a nonholonomic robot are defined from the rolling constraint in (9)

$$\begin{bmatrix}\dot{x} \\ \dot{y} \\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}R\dot{\theta} c\psi \\ R\dot{\theta} s\psi \\ \dot{\psi}\end{bmatrix}. \tag{25}$$

Here, we adopted decoupled controllers for the wheel speed $\dot{\theta}$ and steering angle ψ. The details of the controller are presented in the following subsections.

A. State Estimation

In order to control the robot system, it is crucial to have knowledge of its current state, describing the driving modules as well as the entire robot system. To estimate these states, two IMUs are utilized in each driving module, along with encoder information from the driving motors.

There is an IMU in each driving module measuring the angular velocity, linear acceleration, and absolution orientation of each module. The IMU estimates the absolute orientation with a

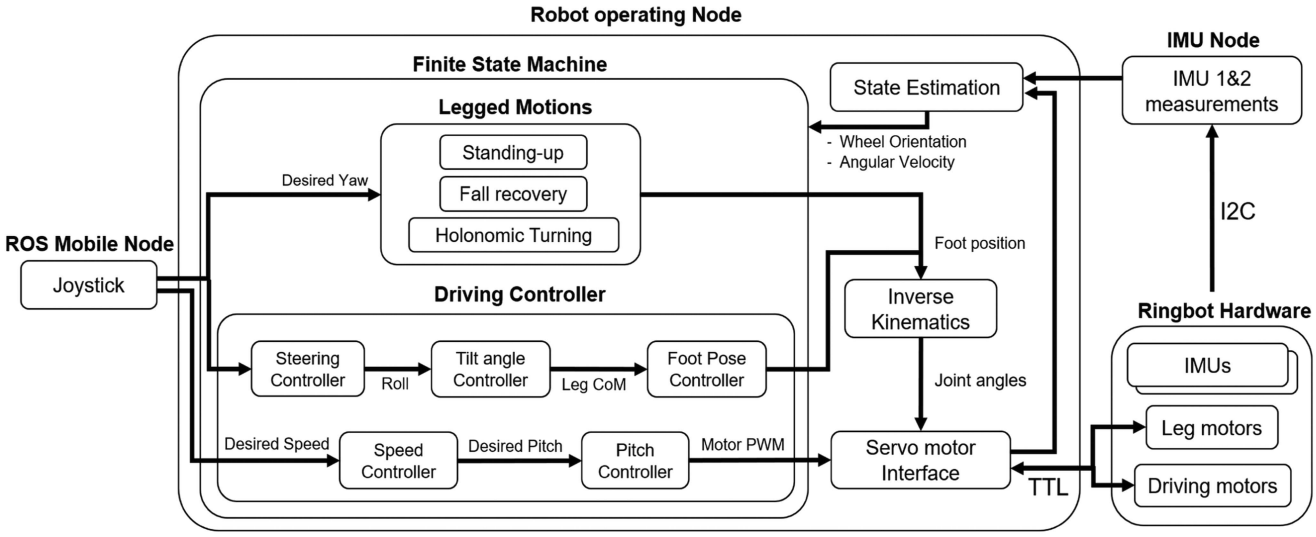


Fig. 7. Schematics for the software implementation of Ringbot.

100 Hz rate using an internal sensor fusion algorithm to combine data from the gyroscope, accelerometer, and magnetometer.

The yaw and roll orientation of the wheel are estimated from the IMU values. The IMU value output for the absolute orientation is obtained in a quaternion. By converting the quaternion into a 3-1-2 Euler angle, the roll and yaw orientation of the wheel frame can be estimated, and then the pitch orientation for describing the driving module's movement can be obtained. This is because the wheel frame undergoes a 3-1 Euler angle sequence transformation from the world coordinate, as defined in Section III.

The acquisition of absolute roll and yaw angles from the IMUs on each driving module provides a redundant measure of the wheel's roll and yaw orientation. Leveraging this redundancy enhances the accuracy of estimating the wheel's yaw and roll angles. This is achieved by averaging the measurements, which reduces noise and drift that might occur with a single IMU. In addition, this method helps to minimize bias arising from hardware-related errors, such as inaccuracies during installation or slack in the rails.

Consequently, when the quaternion value $[w_i, x_i, y_i, z_i]$ represents the absolute orientation measured by the IMU at driving module i, the estimated roll and yaw angles can be derived as follows:

$$\hat{\psi} = \frac{1}{2}\left(\text{atan}\left(\frac{-2(xy_1 - wz_1)}{1-2(x_1^2+z_1^2)}\right) + \text{atan}\left(\frac{-2(xy_2 - wz_2)}{1-2(x_2^2+z_2^2)}\right)\right)$$

$$\hat{\phi} = \frac{1}{2}\left(\text{asin}(2(yz_1 + wx_1)) + \text{asin}(2(yz_2 + wx_2))\right).$$

Since the driving modules only rotate along the y axis of the wheel, the pitch orientation of IMU can be directly considered as θ_{1_1} and θ_{1_2} in Fig. 6(b).

$$\hat{\theta}_{1_i} = \frac{\text{asin}(-2(xz_i - wy_i))}{\cos\phi_i}$$

where the subscript i indicates the specific driving module.

The wheel's angular velocity is also estimated using the IMUs, which are used for the roll angle control, driving velocity control, and steering control. Since each driving module only has a pitch angle rotation relative to the y-axis of the wheel frame, the wheel's angular velocity can be obtained by projecting the measured angular velocity vectors $[\omega_{x_i}, \omega_{y_i}, \omega_{z_i}]^T$ to the wheel frame axes. As the wheel's angular velocity can be obtained from each IMU, the two values are averaged to estimate the angular velocity of the wheel frame less affected by sensor noise and bias

$$\begin{bmatrix}\hat{\omega}_{x_w}\\ \hat{\omega}_{y_w}\\ \hat{\omega}_{z_w}\end{bmatrix} = \frac{1}{2}\begin{bmatrix}\omega_{x_1}c\hat{\theta}_{1_1} - \omega_{z_1}s\hat{\theta}_{1_1}\\ 0\\ \omega_{z_1}c\hat{\theta}_{1_1} + \omega_{x_1}s\hat{\theta}_{1_1}\end{bmatrix} + \frac{1}{2}\begin{bmatrix}\omega_{x_1}c\hat{\theta}_{1_1} - \omega_{z_1}s\hat{\theta}_{1_1}\\ 0\\ \omega_{z_1}c\hat{\theta}_{1_1} + \omega_{x_1}s\hat{\theta}_{1_1}\end{bmatrix}.$$

Similar to the orientation estimation, the y-axis angular velocity of the wheel is zero. Instead, we consider ω_{y_1} and ω_{y_2} as the angular velocities corresponding to each driving module's respective joint, represented by $\dot{\theta}_{1_1}$ and $\dot{\theta}_{1_2}$.

The relative position between two driving modules, θ_{dist}, is estimated through the driving motor encoder measurements, which are more reliable than IMUs measurements. Since the distance between the driving modules is strictly regulated by a feedback controller with a large feedback gain, it is crucial to minimize sensor errors like noise or delay.

The driving motor has an absolute encoder that can measure a motor output angle with 0.088° resolution, with up to 256 revolutions in both clockwise and counterclockwise rotation. Note that upon rebooting the motor system, the stored encoder value is erased and initialized in a range of 0° to 360°, even if it previously surpassed this range before the reboot.

Due to the encoder values reset, an initial calibration is necessary using the pitch angle obtained from the IMU measurement. Once the initial IMU measurement and initial motor position sensor values are captured, the relative angle between the two modules, θ_{dist}, is estimated as follows:

$$\hat{\theta}_{dist} = \left(\hat{\theta}_{1_{1_0}} - \hat{\theta}_{1_{2_0}}\right) + N_w N_d \left((\theta_{m_1} - \theta_{m_{1_0}}) - (\theta_{m_2} - \theta_{m_{2_0}})\right).$$

Here, θ_{m_1} and θ_{m_2} represent the motor position sensor values, while N_w and N_d denote the gear ratio. The subscript 0 indicates that the value is an initial value.

B. Speed Controller

The speed controller's design is based on the intuitive principle that an offset in the CoM produces torque, propelling the wheel toward the biased direction to recenter the CoM. If the lumped CoM of both driving modules shifts forward, the wheel accelerates, and conversely, it decelerates if the CoM shifts backward. Consequently, speed control can be accomplished by simply increasing or decreasing the driving module motors' speed.

Nevertheless, controlling the relative position between the two driving modules during driving is also crucial, as an internal collision between them could lead to hardware issues. Moreover, if the gap continues to widen, the connecting wire between the modules could become overstretched and disconnected. Asynchronized movement also hinders the efficiency of shifting the CoM. To maintain the distance between the driving modules while concurrently managing wheel speed, the controller depicted in Fig. 8(a) is employed.

This controller features an outer angular speed feedback loop, an inner orientation feedback loop, and an additional train distance feedback loop. The design of the controller is designed on a very intuitive behavior of the monocycle: To accelerate the wheel's rotation, increased torque is necessary. To generate more torque, the driving modules need to climb up the ring, thereby utilizing their mass more effectively for the driving torque. For the first part, there is a speed feedback controller, and the output of this controller then feeds into the pitch orientation controller, integrating these two aspects. As the driving modules move within the wheel along a circular rail, their ascent corresponds to larger pitch orientations. To ensure that a driving module does not surpass the wheel's peak point, we cap the reference pitch angle at 180°, preventing the modules from exceeding the peak.

As shown in Fig. 6, the output from the outer speed control loop, comprising a PD controller and a command feedforward, is utilized to set the pitch orientation for each module, corresponding to θ_{1_1} and θ_{1_2}. This approach is preferred over directly manipulating motor PWM with the speed controller output.

The output from the speed controller is added to the initial pitch position of each driving module and is then used as an input for the pitch orientation controller of each respective driving module. The pitch orientation controller output is converted to the driving module motors' PWM signal, controlling the speed of the motors. In addition, a PI controller regulates the distance between the two modules at a desired distance by superimposing its output onto the pitch orientation controller output.

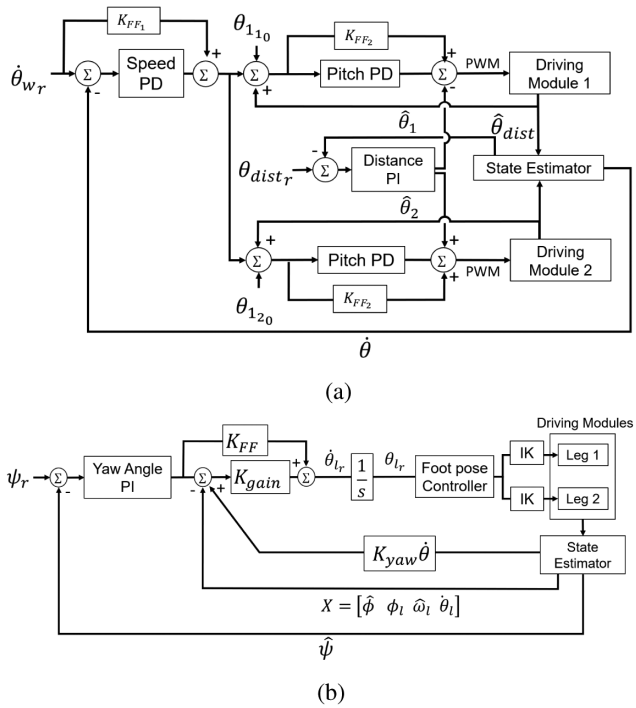


Fig. 8. Decoupled controller of Ringbot. (a) Wheel speed controller. (b) Steering angle controller.

Consequently, the proposed speed controller can control the speed of the wheel while regulating the distance between two driving modules at the same time.

C. Steering Angle Controller

In a unicycle configuration with a single wheel, it is not possible to change the heading direction when the wheel is stationary. Steering of the wheel can result from gyroscopic precession, which occurs when the wheel rolls with a nonzero angular velocity and an external force, such as gravity, causes a change in the rotation axis perpendicular to both the angular momentum and the external torque. Therefore, a roll angle change can cause steering of the rolling wheel. As the direction of the precession is determined by the right-hand rule, it matches with the direction of the cross product of the angular momentum L and the torque τ.

Fig. 9 illustrates the turning direction based on the interplay between the angular momentum and gravitational torque. It is clear that the wheel's turning direction is towards the roll angle deviation, regardless of whether it is moving forward or backward.

Using the correlations between the precession rate and roll angle, a steering angle controller is implemented that consists of an outer loop for the yaw angle control and an inner loop for the roll angle loop. The outer loop includes a PI controller to track the desired yaw angle of the wheel. It is challenging to achieve effective yaw angle control solely with the feedback controller

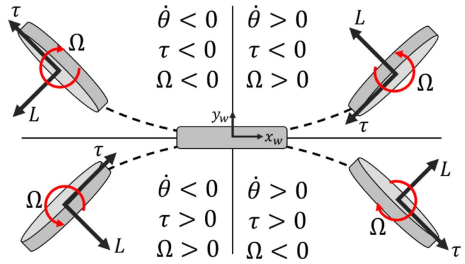


Fig. 9. Turning direction of the rolling wheel caused by the procession effect depends on the rolling velocity and roll angle.

as the yaw angle dynamics as complexly coupled with the other states of the robot, particularly the rolling angular velocity. The equation for a precession rate is given as follows:

$$\Omega = \frac{\tau}{L}. \tag{26}$$

The equation illustrates that the precession rate is inversely related to the angular momentum, denoted by $L = I\dot{\theta}$. Therefore, the same feedback controller gain might not work well for different driving speeds. In order to address this problem, a compensation term $K_y a w \dot{\theta}$ is applied to compensate for the influence of the angular velocity.

The steering angle controller's output serves as an input for the roll angle controller. Roll angle control of the wheel is accomplished by shifting the legs laterally as a 2-DoF inverted pendulum. As described in Section III-C, the simplified lateral dynamic model represents the wheel as the first linkage connected to the ground with a revolute joint at the contact point. The legs are then considered as the second linkage connected. When the wheel is not rolling, there is no torque on the first joint, which makes the model equivalent to the Acrobot model. The Acrobot, an underactuated 2-DoF system, has been a prominent subject of nonlinear control research [24], [25], [26], [27]. A widely recognized method for Acrobot balancing control is employing a state feedback control law $U = -KX$ for the second joint torque with linear quadratic regulator (LQR) gains [23]. Ringbot adopts a similar state feedback control law to control the wheel's roll angle. The control states are defined as

$$X = [\phi \quad \phi_l \quad \dot{\phi} \quad \dot{\phi}_l]^T. \tag{27}$$

It is important to recognize that ϕ_l corresponds to the joint angle of the second link in the simplified lateral dynamic model depicted in Fig. 4, representing the lumped model of the two legs. When the robot is in driving mode, the legs only make lateral movements through the third joint (θ_{3_i}) of the leg, consistently maintaining a configuration where each leg is the mirror image of the other, therefore the legs can behave as one inverted pendulum with a fixed length. This also prevents internal collisions between the legs. Unlike the conventional Acrobot controller, Ringbot's roll angle controller regulates the joint velocity $\dot{\phi}_l$ instead of the joint torque, as the leg servo motors are operated in a position control mode. As the legs are also involved in diverse legged motion tasks, it is more beneficial to use the motor in position control mode and take advantage of the strong position control capabilities offered by the servo motor's built-in controller. Therefore, the control effort needed to be converted into a position command to the motors.

The control loop rate of the system is limited to the relatively low rate of 100 Hz due to the motor communication speed and the IMU sampling rate. If the control effort is treated as the joint acceleration and integrated twice to calculate corresponding the joint position, it causes significant delay and lagging due to the slow discrete loop time. Therefore, we decided to integrate the control effort only once to control the joint velocity instead.

As a result, the state feedback control law in the ith discrete step is

$$U = \dot{\theta}_{l_{r_i}} = -KX_i.$$

The control effort is then numerically integrated using the trapezoidal rule to compute the corresponding joint angle.

$$\theta_{l_{r_i}} = \theta_{l_{r_{i-1}}} + 0.5\left(\dot{\theta}_{l_{r_i}} + \dot{\theta}_{l_{r_{i-1}}}\right)\Delta T.$$

Here, ΔT stands for the loop time. Given that the legs move laterally in unison, functioning as a single inverted pendulum with a fixed length, the position of each foot can be determined in a way that aligns with the CoM position defined by θ_l and the pendulum length l_{c_2}. Once the foot position for each leg is established, the joint angles of the legs can be calculated using inverse kinematics.

When the wheel is stationary, the robot can be considered as an underactuated system due to the absence of torque at the first joint in the simplified model. Consequently, the control state can only be manipulated around the fixed upright equilibrium point ($\phi = \phi_l = 0.$). Although the wheel features a flat surface, allowing for a substantial margin to maintain an upright stance without control, balancing control is still essential to maintain balance for uneven ground surfaces and external disturbance.

As the wheel moves with adequate rolling speed, the gyroscopic torque stabilizes the robot by counteracting the torque induced by gravity. Hence, the same controller's capability extends beyond the fixed point, allowing it to manage the wheel's nonzero roll angle. Since the state feedback gains are tuned for the static balancing control case, a feedforward term is incorporated to track the desired roll angle more effectively without excessively increasing the gain, which could potentially make the system unstable.

D. Legged Motion

Ringbot utilizes its legs for multiple purposes, not only for balancing and steering, but also actively engaging with its surroundings to assist the wheeled driving. A finite state machine (FSM) has been developed to facilitate the transition between driving mode and various legged motion modes, as illustrated in Fig. 11. Transitions between states are prompted by specific conditions indicated alongside the arrows. Detailed explanations for each state are provided below as follows.

1) Driving Mode: In driving mode, the speed and steering angle controllers work together to follow user commands or a predefined trajectory. This mode serves as Ringbot's default setting and can also be considered a standby mode without the

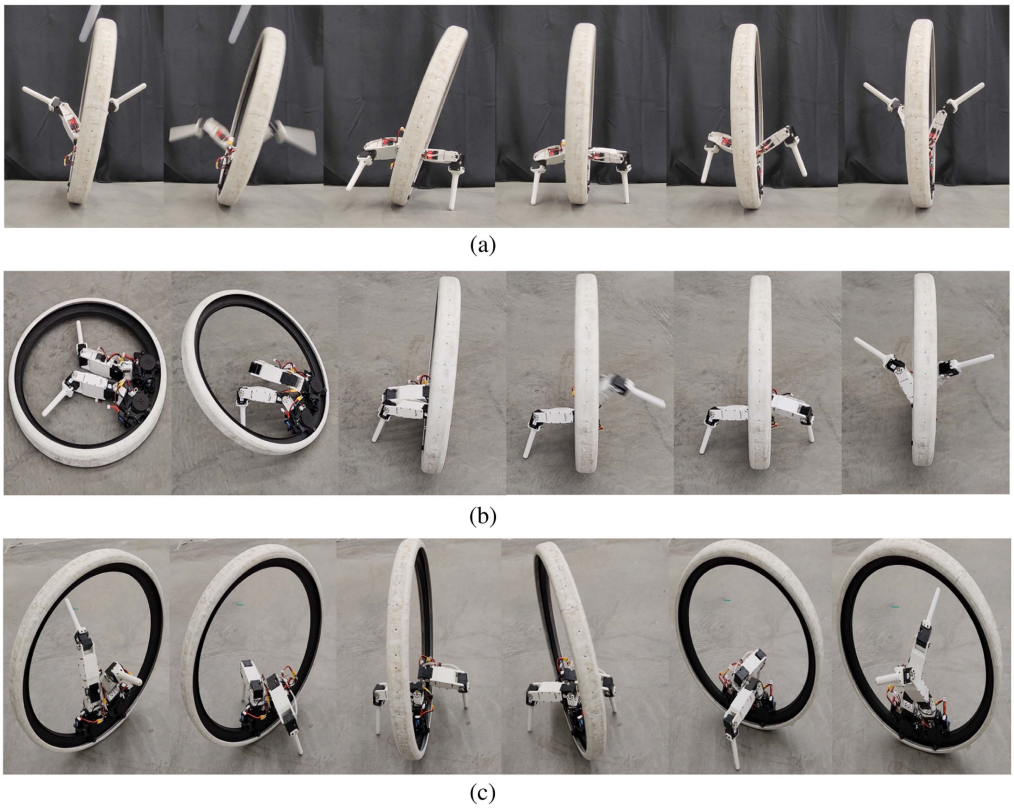


Fig. 10. Snapshots of the implemented legged motions. The legged motions are also demonstrated in the supplementary video. (a) Fall recovery motion. (b) Standing-up motion. (c) Holonomic turning motion.

commands. When the wheel remains stationary, the legs are controlled to maintain a stable, upright posture.

2) Fall Recovery Mode: In this mode, the system stabilizes the robot's falling body by stepping on the ground and pushing back to return the wheel to an upright position. The mode is activated when the wheel is stationary, and the roll angle deviation exceeds 15°. Under these conditions, the robot is deemed to be losing balance and falling. Upon activation, the robot's feet extend sideways and make contact with the ground by estimating its height based on the roll angle measurement. After the feet make contact, the supporting leg on one side lifts the robot's body to restore its upright position. Subsequently, the feet are raised, and the mode transitions back to the driving mode when both the roll angle and the rolling angular velocity are zero. The demonstration of fall recovery motion is presented in Fig. 10(a).

3) Standing-Up Mode: Fig. 10(b) shows the snapshots of the standing-up motion. When the roll angle deviation exceeds 60° while the wheel's linear velocity is zero, the system assumes the wheel falls down on the ground and transitions to the standing-up mode rather than Fall Recovery mode. This is to avoid potential damage to the robot's structure and leg motors, which might occur from the increased load of lifting the robot's weight at a larger roll angle. When the mode is activated, it waits for 1 s, allowing the robot to come to a complete stop and rest on the ground. Subsequently, the robot's body is lifted by utilizing both legs while using the contact point between the wheel and the ground as a pivot. As the roll angle reaches zero, the leg of driving module 2 rapidly moves to the opposite side of the wheel and steps on the ground to prevent the wheel from falling in the other direction. Once the wheel is stabilized, the feet are raised, and the system reverts to driving mode.

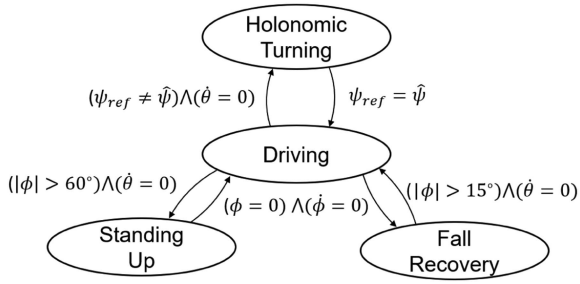


Fig. 11. Visual representation of the leg motion FSM.

4) Holonomic Turning Mode: This mode changes the wheel's yaw angle through leg motion holonomically. The non-holonomic monocycle cannot change its heading direction while stationary. Therefore, if a user commands a yaw angle change while the wheel's velocity is zero, it becomes impossible to follow the instruction. To address this, the holonomic turning mode is activated, utilizing the legs to make a step and rotate the robot in place, allowing it to track the yaw angle command. The demonstration of the turning motion is presented in Fig. 10(c).

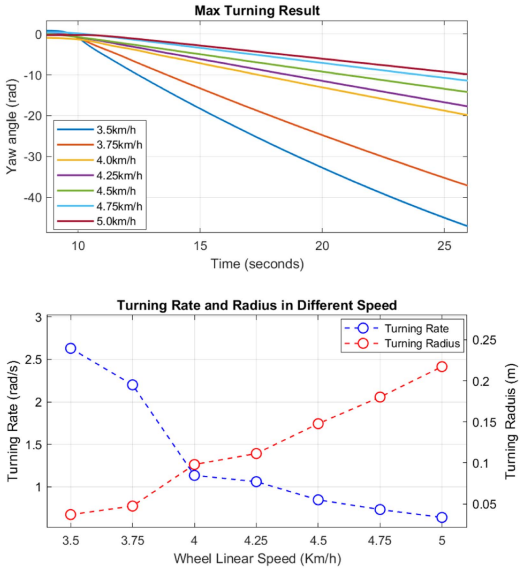


Fig. 12. Simulation results for identifying the maximum turning rate.

VI. Simulation

A multibody dynamics simulation is utilized to verify hardware design and controller performance. The simulation model was developed using MATLAB 2022a Simscape Multibody. Accurate mass properties obtained from the CAD and fabricated hardware prototypes were reflected in the simulation model.

A. Steering Rate

Through the simulation, we assessed Ringbot's steering capacity. In the simulation scenario, Ringbot maintained a steady linear velocity while its legs were shifted to one side until the maximum range of motion was reached for getting the maximum turning rate. Simulations were conducted at velocities from 3.5 km/h, the slowest linear velocity, where the simulation model was able to sustain balance with the weight shift, to 5 km/h, the robot's fastest linear velocity. The results are shown in Fig. 12.

As indicated by (26), the precession rate is proportional to the external torque and inversely related to the angular momentum. The robot has more substantial angular momentum when the angular velocity is higher. Meanwhile, the external torque is decreased, as the greater gyroscopic torque in (22) has a greater effect, reducing the roll angle deviation caused by the same leg shifting. Thus, the wheel's turning rate decreases as the linear speed rises. The simulation outcomes correspond with the previously mentioned phenomenon, where turning occurs at a slower rate when the speed is higher. At 5 km/h, the simulation model successfully executed a turn with a 0.217-m turning radius while it made the sharpest turning with 0.037 m at 3.5 km/h.

B. Driving Performance

The decoupled controllers for speed and steering were validated through a simulation in which the model simultaneously tracked both speed and steering angle commands. The controllers proposed in Section V were used to control the simulation model. The plots in Fig. 13 present the results of the simulation tracking varying speed commands on a straight trajectory without steering. The speed commands increment by 1 km/h every 10 s, reaching a peak at 5 km/h, the maximum speed. This speed is maintained for 20 s, followed by a rapid decrease in velocity to 0, simulating an aggressive braking scenario. To assess mechanical configurations, such as motor selection, gear ratio, and wheel size, both the speed and torque of the driving motor were monitored. The results illustrate the simulation model's adeptness in adhering to diverse velocity commands and efficiently transitioning from maximum velocity to a complete halt in just 2.6 s, all without exceeding the operational limits of the driving motor.

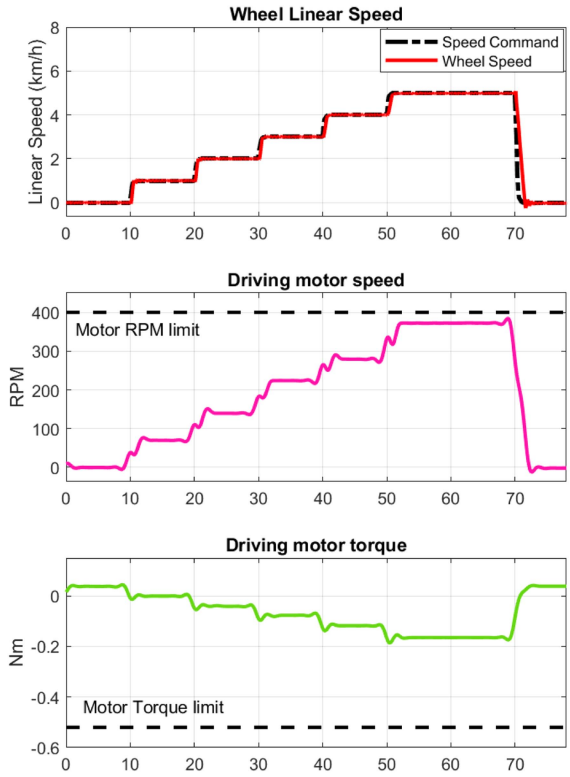


Fig. 13. Simulation result for varying speed commands and braking.

An additional simulation was carried out to evaluate the driving performance during steering, wherein the robot maintained a constant speed of 5 km/h while following arbitrary desired yaw angle commands set by a pilot. Motor torque and speed saturation were also incorporated into the simulation model to confirm the hardware design and specifications through the simulation. As depicted in Fig. 14, the robot successfully maintained a constant speed while steering to follow the arbitrary yaw angle inputs.

VII. Hardware Experiment

The hardware experiment was conducted using the fabricated robot hardware. A pilot manipulated the speed and steering commands via a joystick user interface implemented in an Android mobile application, controlling the Ringbot hardware

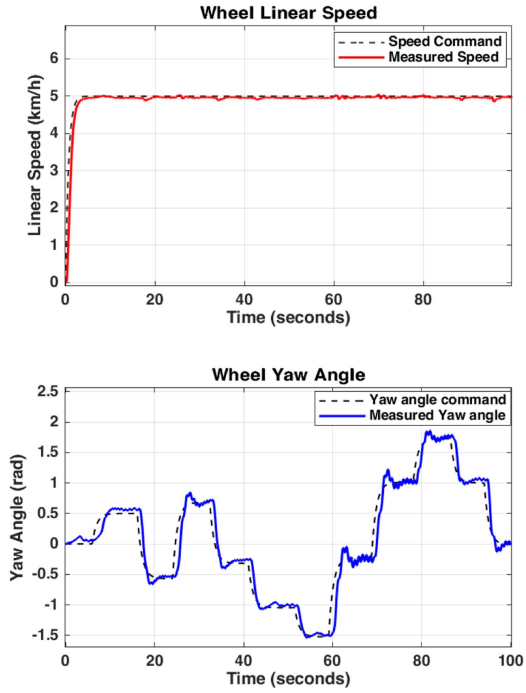


Fig. 14. Simulation result for tracking arbitrary steering angle inputs at a constant speed command.

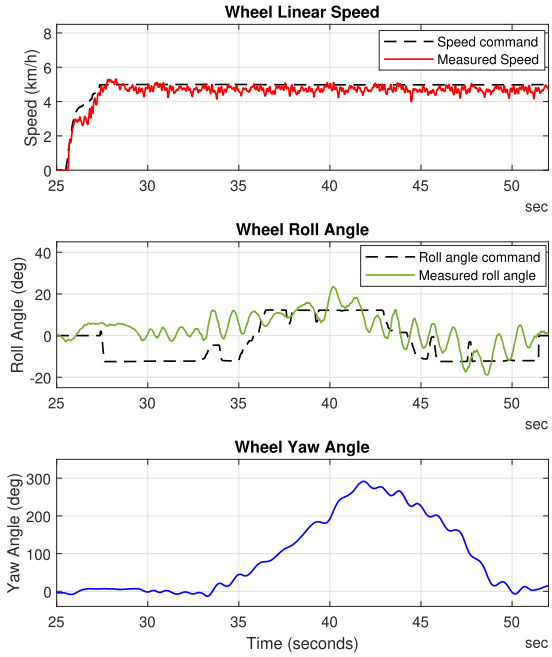


Fig. 15. Hardware experiment results for tracking steering command at a constant speed command.

remotely. Similar to the simulation setup, the speed command was gradually increased until the robot reached its maximum linear speed of 5 km/h.

For the driving test, the hardware prototype was driven in 8 m $\times$ 8 m space by controlling the speed and steering angle through a remote control by a human pilot. For the experiment, we used only the inner loop of the steering controller, which shifts the legs to control the wheel's roll angle. This setup allowed the pilot to intuitively and responsively control the heading direction by directly shifting the legs based on the user's steering command.

The results, displayed in Fig. 15, indicate that the hardware was able to track the speed command and achieved the maximum speed, as seen in the simulation. Even though the roll angle did not precisely follow the desired command due to abrupt changes and short settling times to drive the robot in a limited space, the yaw angle change was smoothly adjusted according to the user's command. The demonstration of the driving experiment can be found in the supplementary video.

The current Ringbot hardware is designed as a smaller-scale prototype to illustrate the feasibility of our novel monocycle-with-legs concept. Looking ahead, the model intended for practical applications is expected to have around two to three times larger wheel size than that of the existing prototype.

With this potential increase in size, it becomes crucial to ascertain whether the larger iteration could navigate real-world road obstacles, such as speed bumps or curbs. According to existing guidelines, the standard heights for curbs and speed bumps are reported to be 140 mm [28] and 75–100 mm [29], respectively. In terms of the larger robot's wheel diameter, these measurements correspond to approximately 10%–14% for curbs and 6.6%–7.7% for speed bumps. To assess whether the larger robot would have sufficient ability to traverse these obstacles, experiments were performed with the current prototype.

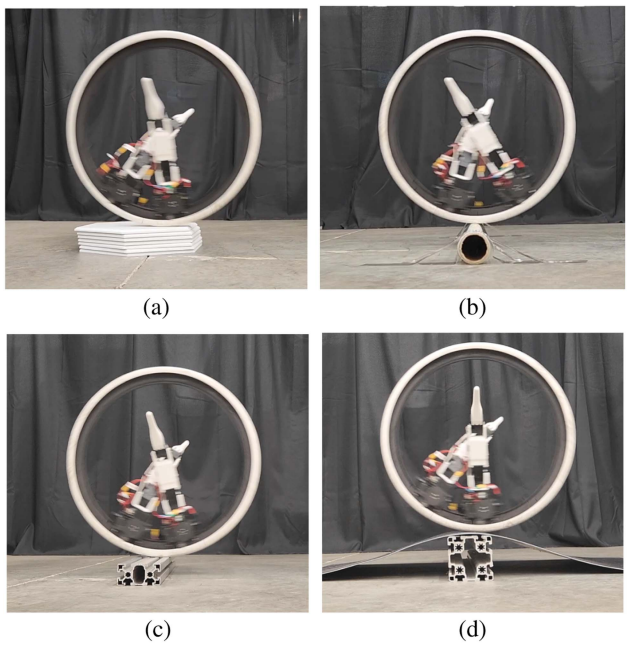


Fig. 16. Ringbot goes over obstacles with various shapes and heights. (a) Seven stacks of hexagonal acoustic panels. (70 mm) (b) a cylindrical pipe. (60 mm) (c) T-slotted aluminum frame. (45 mm) (d) T-slotted aluminum frame with a rubber mat. (90 mm)

Thanks to its large wheel diameter, the Ringbot displayed an excellent capacity to navigate over various shapes and sizes of obstructions in the wheeled driving mode, as depicted in Fig. 16.

To assess the robot's capability, we conducted tests using acoustic panels measuring 355 mm × 304 mm × 10 mm each, and the difficulty level was gradually escalated by increasing the number of stacked panels in each round of testing. The robot was able to surmount up to seven stacked panels, a height of 70 mm. Furthermore, it smoothly navigated over cylindrical obstacles such as a 60-mm pipe and a 45-mm rectangular aluminum frame. In the most extreme test scenario, the robot was able to traverse and maintain balance on a simulated speed bump with a height equivalent to 17.5% of its wheel diameter, as depicted in Fig. 16(d).

This success in obstacle navigation for the smaller-scale prototype provides promising implications for the larger, practical application version's ability to manage real-world road obstacles.

VIII. Conclusion

In this article, we introduced the mechanical design and control framework of a novel wheel-leg robot with legs named Ringbot. The primary goal of this study is to create a prototype robot that validates the new robot mechanism concept. Our efforts are directed towards not only developing the hardware, but also designing controllers, emphasizing the effectiveness of a monocycle robot with legs.

By adopting a monocycle mechanism, the robot benefits from inherent pitch stability and excellent obstacle-overcoming capability. In addition, the large ring-shaped wheel provides ample internal space for multi-DoF legs. The driving components and a driver in the traditional monocycle vehicle were replaced with two compact driving modules, each equipped with a leg. For single-wheel driving, a decoupled controller was introduced that controls the pitch angle of the driving modules for speed and controls the lateral shift of the legs for steering.

Ringbot employs its legs for multiple functions, including maintaining balance and direction control in its wheeled mode, as well as executing diverse legged movements that assist its wheeled maneuvering. The legs enable Ringbot to prevent falling and even to self-right itself autonomously. In addition, Ringbot can holonomically turn its heading direction using the legs.

Experiments involving both simulation models and physical hardware were conducted to evaluate the robot's capabilities and confirm the design's soundness. The prototype demonstrated efficient driving performance in simulations and with actual hardware, adeptly tracking a variety of speed and steering commands set by a pilot. It also showcased the ability to navigate over different obstacles, benefiting from its large wheel size. However, the testing was limited to flat surfaces with minimal incline, indicating that the robot's performance on uneven terrain or ramps remains untested and may show limitations under such conditions.

Nevertheless, the primary goal of the two-legged Ringbot is to establish the feasibility and effectiveness of this novel mechanism, laying a solid foundation for both hardware and software implementation. Although the current two-legged version primarily uses its legs to assist in driving, future developments will focus on a four-legged Ringbot. This evolution is aimed at enhancing its ability to handle challenging terrains by seamlessly switching between driving and full-legged locomotion, thereby offering a versatile solution for diverse environments.

This work showed the implementation and strengths of the legged monocycle mechanism. The demonstrated prototype, designed at a small scale, serves to validate the concept of a Ringbot. Yet, it also highlighted the possibility of an enlarged version that could be effectively implemented as small-footprint autonomous transportation for city and indoor environments, capitalizing on its versatile navigation abilities.

One particularly promising application for the Ringbot is in last-mile delivery service, where high versatility and maneuverability are crucial for navigating various terrains including paved roads, sidewalks, stairs, and indoor environments. The ability of the Ringbot to shift its locomotion modes according to different situations and environments equips it to adeptly handle a range of terrain conditions. Moreover, its compact footprint enhances its suitability for urban areas and indoor settings, making it an even more attractive solution for such applications.

In our future work, we aim to further develop the Ringbot concept into a four-legged configuration, developing a fully transformable wheel-leg robot capable of both quadruped walking and wheeled driving. This endeavor will involve not only hardware improvements, but also the development of appropriate quadrupedal movements and smooth transitions between various locomotion modes. In addition, the experiment results have highlighted the impact of driving speed on turning capabilities, highlighting the need for a smart strategy that manages both speed and steering while considering the dynamics and upcoming path.

References

[1] H. Bae, I. Lee, T. Jung, and J.-H. Oh, "Walking-wheeling dual mode strategy for humanoid robot, DRC-HUBO," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 1342–1348.

[2] A. Stentz et al., "CHIMP, the CMU highly intelligent mobile platform," *J. Field Robot.*, vol. 32, no. 2, pp. 209–228, 2015.

[3] S. Karumanchi et al., "Team RoboSimian: Semi-autonomous mobile manipulation at the 2015 DARPA robotics challenge finals," *J. Field Robot.*, vol. 34, no. 2, pp. 305–332, 2017.

[4] M. Geilinger, R. Poranne, R. Desai, B. Thomaszewski, and S. Coros, "Skaterbots: Optimization-based design and motion synthesis for robotic creatures with legs and wheels," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–12, 2018.

[5] M. Bjelonic et al., "Keep Rollin'—Whole-body motion control and planning for wheeled quadrupedal robots," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 2116–2123, Apr. 2019.

[6] M. Bjelonic, P. K. Sankar, C. D. Bellicoso, H. Vallery, and M. Hutter, "Rolling in the deep–hybrid locomotion for wheeled-legged robots using online trajectory optimization," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 3626–3633, Apr. 2020.

[7] V. Klemm et al., "LQR-assisted whole-body control of a wheeled bipedal robot with kinematic loops," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 3745–3752, Apr. 2020.

[8] X. Li, H. Zhou, H. Feng, S. Zhang, and Y. Fu, "Design and experiments of a novel hydraulic wheel-legged robot (WLR)," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 3292–3297.

[9] N. Kashiri et al., "CENTAURO: A hybrid locomotion and high power resilient manipulation platform," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1595–1602, Apr. 2019.

[10] S.-H. Yun, J. Park, J. Seo, and Y.-J. Kim, "Development of an agile omnidirectional mobile robot with GRF compensated wheel-leg mechanisms for human environments," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 8301–8308, Oct. 2021.
[11] C. Zheng, S. Sane, K. Lee, V. Kalyanram, and K. Lee, "α-WaLTR: Adaptive wheel-and-leg transformable robot for versatile multiterrain locomotion," *IEEE Trans. Robot.*, vol. 39, no. 2, pp. 941–958, Apr. 2023.
[12] R. Cao, J. Gu, C. Yu, and A. Rosendo, "OmniWheg: An omnidirectional wheel-leg transformable robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 5626–5631.
[13] H.-Y. Wang, L.-J. Chen, W.-S. Yu, and P.-C. Lin, "A wheel to leg transformation strategy in a leg-wheel transformable robot," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, 2023, pp. 293–298.
[14] N. Tan, R. E. Mohan, and K. Elangovan, "Scorpio: A biomimetic reconfigurable rolling–crawling robot," *Int. J. Adv. Robot. Syst.*, vol. 13, no. 5, 2016, Art. no. 1729881416658180.
[15] Z. Huang et al., "Design and analysis of a transformable spherical robot for multi-mode locomotion," in *Proc. IEEE Int. Conf. Mechatron. Automat.*, 2017, pp. 1469–1473.
[16] H.-M. Maus, S. Lipfert, M. Gross, J. Rummel, and A. Seyfarth, "Upright human gait did not provide a major mechanical challenge for our ancestors," *Nature Commun.*, vol. 1, no. 1, 2010, Art. no. 70.
[17] M. J. van den Broek, "Fast self-stable planar bipedal running," Ph.D. dissertation, TU Delft, 2019.
[18] C. W. Bert, "Dynamics and stability of unicycles and monocycles," *Dyn. Stability Syst.*, vol. 5, no. 1, pp. 30–35, 1990.
[19] S.-J. Tsai, E. Ferreira, and C. Paredis, "Control of the gyrover. a single-wheel gyroscopically stabilized robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1999, pp. 179–184.
[20] Y. Ou and Y. Xu, "Stabilization and line tracking of the gyroscopically stabilized robot," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2002, pp. 1753–1758.
[21] P. Cieslak, T. Buratowski, T. Uhl, and M. Giergiel, "The mono-wheel robot with dynamic stabilisation," *Robot. Auton. Syst.*, vol. 59, no. 9, pp. 611–619, 2011.
[22] M. Forouhar, M. H. Abedin-Nasab, and G. Liu, "Introducing GyroSym: A single-wheel robot," *Int. J. Dyn. Control*, vol. 8, pp. 404–417, 2020.
[23] M. Spong, "The swing up control problem for the Acrobot," *IEEE Control Syst. Mag.*, vol. 15, no. 1, pp. 49–55, Feb. 1995.
[24] S. C. Brown and K. M. Passino, "Intelligent control for an Acrobot," *J. Intell. Robot. Syst.*, vol. 18, pp. 209–248, 1997.
[25] J. Hauser and R. M. Murray, "Nonlinear controllers for non-integrable systems: The acrobot example," in *Proc. Amer. Control Conf.*, 1990, pp. 669–671.
[26] S. Bortoff and M. Spong, "Pseudolinearization of the acrobot using spline functions," in *Proc. IEEE 31st Conf. Decis. Control*, 1992, pp. 593–598.
[27] T. Kobayashi, T. Komine, S. Suzuki, M. Iwase, and K. Furuta, "Swing-up and balancing control of Acrobot," in *Proc. 41st SICE Annu. Conf.*, 2002, pp. 3072–3075.
[28] J. Guerrero, R. Chapuis, R. Aufrére, L. Malaterre, and F. Marmoiton, "Road curb detection using traversable ground segmentation: Application to autonomous shuttle vehicle navigation," in *Proc. IEEE 16th Int. Conf. Control Automat. Robot. Vis.*, 2020, pp. 266–272.
[29] P. A. Weber and J. P. Braaksma, "Towards a North American geometric design standard for speed humps," *ITE J.*, vol. 70, no. 1, pp. 30–39, 2000.

Kevin Genehyub Gim (Member, IEEE) received the B.S. degree in mechanical engineering from Yonsei University, Seoul, South Korea, the M.S. degree in mechanical engineering from the University of California, Los Angeles, CA, USA, and the Ph.D. degree in mechanical engineering from the University of Illinois Urbana-Champaign, Champaign, IL, USA.

His research interests include developing mechanisms for legged robots, biped robots, humanoids, and human–robot interaction.

Joohyung Kim (Member, IEEE) received the BSE and Ph.D. degrees in electrical engineering and computer science (EECS) from Seoul National University, Seoul, South Korea, in 2001 and 2012, respectively.

From 2009 to 2012, he was a Research Staff Member with the Samsung Advanced Institute of Technology, Suwon, South Korea, developing biped walking controllers for humanoid robots. From 2013 to 2019, he was a Research Scientist with Disney Research. Prior to joining Disney, he was a Postdoctoral Fellow with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, for the DARPA Robotics Challenge in 2013. He is currently an Associate Professor of electrical and computer engineering with the University of Illinois Urbana-Champaign, Champaign, IL, USA. His research interests include data/task-driven robot system design and interaction, including limbed robot design optimization, robot motion retargeting and learning, and seamless human–robot interaction.