A differential drive rimless wheel that can move straight and turn

Sebastian Sanchez and Pranav A. Bhounsule

***Abstract*— A rimless wheel, or a wheel without a rim, is the simplest example of a legged robot and is an ideal testbed to understand the mechanics of locomotion. This paper presents the design and control of a differential drive rimless wheel robot that achieves straight-line movement and turning. The robot design comprises of a central axis with two 10-spoked springy rimless wheels on either side and a central body that houses the electronics, motors and transmission, computers, and batteries. To move straight, the current in each motor is servoed to a constant torso pitch angle. To turn, while maintaining constant pitch, a differential current is added and subtracted from either motor. In separate tests, the robot achieved the maximum speed of 9.66 miles per hour, the lowest total cost of transport (power per unit weight per unit velocity) of 0.13, and the smallest turning radius of 0.5 m.**

I. Introduction

A rimless wheel, or a wheel without a rim, is the simplest legged robot. It consists of two rimless wheels connected by a central shaft. Past work has extensively focussed on sagittal plan motion of the rimless wheel limiting its maneuverability to the fore-aft plane. In this work, we present the design and control of a rimless wheel that can turn thus enabling more versatile movement.

When the rimless wheel is pushed with momentum on level ground, the legs of the wheel collide with the ground, losing energy at every step, and eventually coming to a complete stop. Thus, unlike a rimmed wheel which has to bear rolling friction, the rimless wheel has to overcome collisional losses, which are substantial. Thus, to sustain motion, a rimless wheel needs an external power source.

The simplest method of sustaining walking with the rimless wheel is to launch it down a slope. Depending on the mass, inertia, leg length, and the number of spokes, there is a certain slope that ensures steady speed. Such a motion was first analyzed using tools in dynamical systems, namely Poincaré section and limit cycle, by McGeer [1]. The analysis consists of first finding an initial condition at a Poincaré section (a chosen instant in the locomotion cycle such as foot-strike), that repeats at the Poincaré section at the subsequent step, resulting in a limit cycle or periodic motion. Then, using a linearization of the step-to-step dynamics known as the Jacobian, one can determine the stability of the limit cycle. If the largest eigenvalue of the Jacobian is less than 1, which implies that small perturbations would diminish every step, then the system is stable. It is relatively easy to find stable limit cycle for the 2D rimless wheel [2]. However, in 3D, the rimless wheel is stable in the sagittal plane, but not stable in the yaw (heading) direction [3]. One may achieve stable motion by adding a finite width to the rimless wheel, i.e., two rimless wheels connected via a finite width cross bar connecting two rimless wheel [4].

An unpowered rimless wheel can sustain periodic walking on level ground if it reduces impact losses to zero. Gomes and Ahlin [5] created an ingenious design that involved connecting a torsional spring between the legs and the robot frame in such a way that the spring winds-up from mid-stance to support exchange, extracting the energy from the robot to ensure a nearly impact-less collision. Thereafter, the spring restores the energy to the robot by unwinding itself from support exchange to the next mid-stance.

Powering the rimless wheel enables locomotion on flat terrain. Agrawal and Yin [6] created a vehicle with a castor wheel in front and two rimless wheels at the rear. These two rear rimless wheels had one motor to power both the wheels in tandem and another one to expand/contract the leg length relative to each other. The rimless wheel moved straight by rotating both wheels at the same speed but turned by contracting or expanding the spokes of one wheel relative to the other. Laney and Hong [7] created a robot with 2 pairs of rimless wheels, each with 6 telescopic legs. There was a single motor on a diagonal pair of legs that could expand/contract the pair. Thus, there were three motors per rimless wheel. The robot moved straight and turned by retracting/protracting the telescopic legs in a suitable pattern. Most recently, Cotton et al. [8] built a series of rimless wheel robots with two pairs of wheels arranged side by side with fixed-length legs. The rimless wheel moved straight using a single motor on its central axle and turned using another motor that rocks pendulum like articular body in the lateral plane. One robot they built called the Outrunner was 2 feet tall achieved a top speed of 20 mph and another one called HexRunner [9] was 6 feet tall achieved a top speed of 32.2 mph.

Our work builds upon our earlier rimless wheel robot [10], which had two sets of rimless wheels with 8 legs each. A single motor fixed to the axle connected the two rimless wheels propelled the robot forward. However, that robot was not able to turn. Here, we have upgraded the design to include two motors, one for each rimless wheel, to enable turning by spinning them at different speeds. This work details the mechanical, electrical, and controller design. The novelty of the work is the use of a differential drive mechanism, normally used in wheeled robots, for turning. The mechanism

S. Sanchez was with the Department of Mechanical Engineering, The University of Texas at San Antonio, One UTSA Circle, San Antonio, TX 78249, USA. He currently works at Boardwalk Robotics, Pensacola, Florida. His email is sbaz.93@gmail.com

P. A. Bhounsule is with the Department of Mechanical and Industrial Engineering, University of Illinois at Chicago, 842 W. Taylor St., Chicago, IL 60607, USA. Corresponding author email: pranav@uic.edu

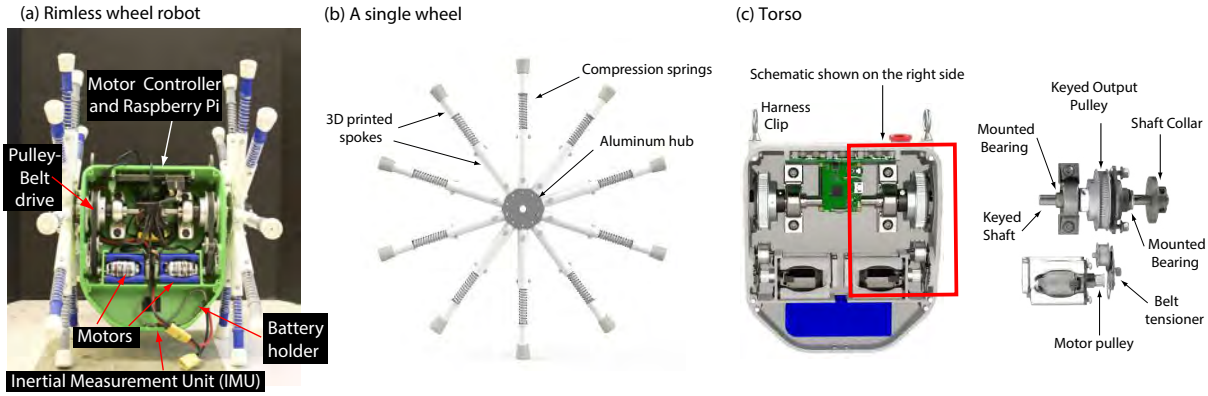

Fig. 1. **Mechanical Design** (a) Rimless wheel photo with important components labelled, (b) a single rimless wheel with compliant spokes, and (c) torso includes the transmission using belt drives, the motors, the sensors, and the batteries.

allows for sharp turns and decouples the sagittal plane motion to lateral tuning. Using this design we demonstrate the best-case energy-efficiency (Total Cost of Transport TCOT, which is total input power per unit weight per unit speed) of 0.13 which is close to that of the most energy efficient legged robot so far, Cargo [11] at 0.11. We also demonstrate a top speed of 9.66 mph in a separate trial and turning radius of 0.5 m. This work does not cover feedback control against external perturbations, which we report elsewhere [12].

The flow of the paper is as follows. The hardware section presents the mechanical and electrical design is in Sec. II, the controller is in Sec. III, the results are in Sec. IV, and discussion in Sec. V.

II. Hardware

The rimless wheel robot called Rowdy Runner 2 (RR2) is an upgraded version of an earlier robot Rowdy Runner (RR) [10]. The most significant additional features in RR2 are its ability to turn, distributed computing, data collection, and remotely operated turning control. All mechanical design files and code is provided online [13].

A. Mechanical Design

We show the Rowdy Runner 2 (RR2) in Fig. 1 (a). The robot comprises two major mechanical components: two sets of rimless wheels (white-colored spokes) placed side-to-side and the torso (green box) in between the wheels.

1) Rimless wheels: The robot has two sets of rimless wheels; we show one of which in Fig. 1 (b). Each set of wheels has 10 legs, each of length 0.26 m. Each leg has four components: a 3D printed tube that attaches to the center hub, an off-the-shelf compression spring, a 3D printed rod that slides into the aforementioned tube, and an off-the-shelf rubber foot. The rod has a lip that contacts the spring and compresses it whenever we place a weight on the leg. A slot designed into the rod allows for constrained movement by securing a screw through the rod and the tube.

All legs connect to a central hub. The hub is one of the highly stressed parts, and hence constructed using aluminum. The hub has 10 small cylinders projecting on the outside, which we used to secure the 10 tubes of the legs through a screw. These screws prevent the tubes from twisting or falling out. We connected each hub of the rimless wheel to the torso (green box) through a shaft using a keyed shaft collar. We clamp the shaft collar onto the shaft by tightening the key which provides a secure connection.

2) Torso: We show the robot torso (green box) in Fig. 1 (c). The torso is 3D printed to allow creating complex geometries such as the bearing alignment holes, circuit board mounts, vent holes, battery compartment, and the motor attachment point.

There are two shafts inside the box that connect to each of the two rimless wheels through a keyed shaft collar. Within the box, each shaft has two mounted bearings; one mounted to the sidewall of the body and one at the end of the shaft inside the body. Having two perpendicularly supported bearings prevents the shaft from moving around from the tension of the belt on the pulleys or during the robot motion.

We fasten the two motors to the wall in the body with the faceplate provided by the manufacturer [14]. Each motor transfers power to the corresponding output shaft through two pulleys and a toothed belt with a 5.4:1 reduction. We attach the encoder to the motor with the help of a 3D printed bracket and the second output shaft. This bracket holds four nuts that thread the screws coming through the wall and the faceplate. We attach the pulley to the output shaft of the motor with two set screws. We align the motor and the output pulley with each other and rotate them by a GT3 toothed belt. The output shaft is a half-inch keyed shaft that engages onto the output pulley. A shaft collar and sidewall bearing surface hold the pulley in place laterally. We install a small tensioner on the sidewall of the body to stretch the belt and prevent the belt from slipping.

We mount all circuit boards directly onto the torso's designed standoffs. When mounting directly into 3D printed plastic, the screws can self-tap themselves into the plastic, eliminating the need for a nut. We mount the motor controller and computer to the top of the body using screws into the designed standoffs. The top of the torso has eyelets for a harness, and a cutout for the computer ports. The

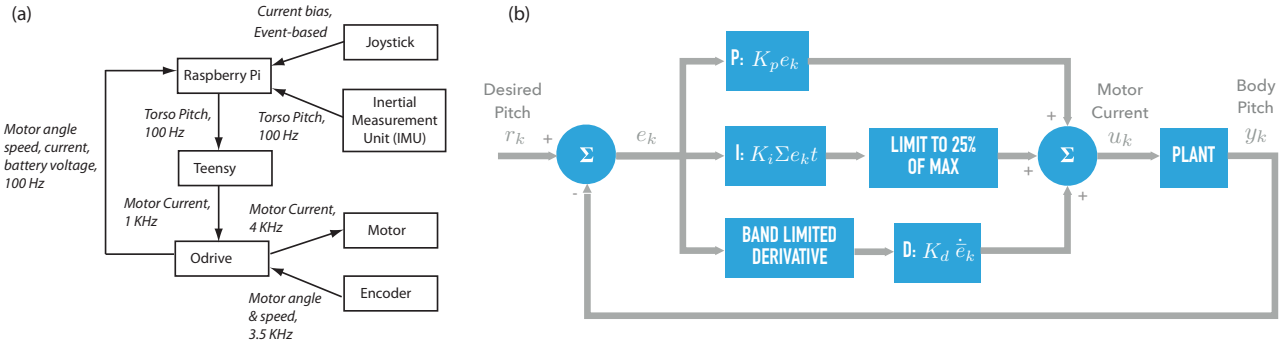

Fig. 2. **Schematic of the electronics layout and feedback controller:** (a) Electronics for distributed control: The Raspberry Pi does all the high-level data management including saving the data and receiving inputs from the Odrive motor controller, the inertial measurement unit, and the joystick. The Teensy does the body pitch control using the body pitch data from the Raspberry Pi. The Odrive regulates the motor current based on the input from the Teensy. (b) Feedback controller: The feedback controller controls the torso pitch using a Proportional-Integral-Derivative (PID) controller at 1000 Hz.

bottom of the torso holds two batteries and the IMU, the latter of which we mount using four screws.

B. Electronics

Figure 2 (a) illustrates the electronics used in the RR2 robot. At the highest level is the Raspberry Pi 3B (4 core, 1.2 GHz, 1 GB RAM) [15] that is responsible for system scheduling, reading and storing all sensor data, and relaying data to an external computer. At the middle level is a Teensy 3.2 micro-controller (Cortex-M4, 72 MHz, 64 KB RAM) [16] that uses the body pitch measurement to compute the desired motor current for body torso control. We connect it to the Raspberry Pi through an USB connection. At the lowest level is an Odrive V3.5 [17], a motor controller for brushless direct current (BLDC) motors. We connect the Odrive to the Raspberry Pi through USB and to the Teensy through a serial connection.

The RR2 uses two commercially available BLDC motors with a 280 KV rating [14]. A 280 KV motor will produce 1 V across its terminals if the motor spins at 280 rpm. In general, a high KV rating is preferred for high speed, low torque applications, but a low KV rating is preferred for high torque, low speed applications. We fit each motor with CUI AMT102 quadrature encoder. The Odrive counts the encoder tics and controls the BLDC motor. We mount a 9 Degree Of Freedom (DOF) orientation sensor BNO055 from Bosch on the torso and is used to measure the torso pitch. We connect this orientation sensor to the Raspberry Pi through a serial connection. The ODrive relays the motor angle, speed, current, and battery voltage to the Raspberry Pi. We use a Dualshock 3 Controller as a remote joystick to control the robot. This joystick has a Bluetooth transmitter, which we pair with the Raspberry Pi. Because of the Raspberry Pi's mediocre internal WiFi antenna range and performance, we attach a small wireless hotspot to the robot torso for communication and data transfer. We connect this hotspot to the Raspberry Pi through an Ethernet cable which we power using an USB battery bank (5000 mAh).

We use a 6S 30C 3000 mAh Lithium Polymer (LiPo) battery with a nominal battery voltage of 22.2 V to power the motors. The 30C rating designates that this battery can output a constant current of 90 Amps (A). The 6S rating means that this battery has 6 battery cells in series, totaling the 22.2 V for nominal voltage. We use a double cell 3.7V 4000 mAh LiPo battery to power the Raspberry Pi. We connect the battery to a circuit board soldered to the Raspberry Pi that boosts the battery voltage to 5 V and prevents the under-voltage of the battery.

The various electronics communicate at different rates. At the highest level, the Raspberry Pi communicates with the inertial measurement unit, Teensy, and Odrive at 100 Hz. At the middle level, the Teensy communicates with the Odrive at 1 kHz. The Odrive communicates with the encoder at 3.5 kHz and with the motor at 4 kHz. The joystick is an exception that communicates with the Raspberry Pi in an event-driven fashion, i.e., the Raspberry Pi accepts the commands from the joystick only at certain times (e.g., it accepts turning command once-per-step at mid-step).

III. Controller

A. Torso Pitch Controller

The goal of the torso pitch controller is to maintain the torso to a set angle $r_k = \phi_{\text{ref}}$ to the vertically downward direction. By holding the torso at an angle to gravity, it torques the leg in contact with the ground. This produces a traction force at the spoke contact that pushes the system forward, thus adding energy into the system. However, the robot loses energy when the legs collide with the ground and friction in the springs and bearings. For a constant torso pitch angle, the energy added by the torso equals the energy loss, leading to steady-state speed for the rimless wheel.

A proportional integral derivative (PID) controller is used to achieve torso pitch control. Fig. 2 (b) shows the control diagram of the PID controller. The PID controller at time step k is given by

$$u_k = K_p e_k + K_i \sum e_k + K_d \dot{\bar{e}}_k \tag{1}$$

where u_k is the current; K_p, K_i, and K_d are the proportional, integral, and derivative gains respectively and are all constant during all runs; $e_k = r_k - y_k$ is the position error and given

by the reference pitch angle r_k and the measured pitch angle y_k respectively. We set the reference pitch angle to $r_k = 50^\circ$ unless noted otherwise. To prevent integral windup, we limit the integral term ($K_i \sum e_k$) to 25% of the maximum current allowed. The error rate term $\dot{e}_k = e_k - e_{k-1}$ is noisy, we use an exponential filter to find the filtered error rate $\bar{e}_k = \alpha e_k + (1-\alpha)\bar{e}_{k-1}$. We initialize the filtered error rate to $\bar{e}_1 = e_1$ and we experimentally tune the constant parameter $\alpha = 0.1$.

To tune the PID gains, we place the robot on a table with the legs firmly clamped. We use the heuristics proposed by Wescott [18] as follows. All gains start at zero and we set K_p at an arbitrary number between 0 and 1. The starting value for K_d is $100 \times K_p$, and we increase K_d until we see excessive oscillation or overshoots from the torso. We then reduce the almost unstable K_d by a factor of 2 to ensure a well-behaved gain. Next, we tune K_p by starting with a value between 1 and 100. We tune the K_p gain to where the body oscillates and then fine-tuned by increasing/decreasing by a factor of 2. Finally, we tune K_i by setting the gain to a value between 0.0001 and 0.01. We then fine tune the value using the techniques for the previous gains. Using this method, our tuned gains were $K_p = 0.17$ A/rad, $K_i = 0.0005$ A/rad-s, and $K_d = 16$ As/rad.

B. Straight motion

To move straight, both the rimless wheels need to move at the same speed. We achieve this by setting the current in both the motors to the same value. Also, we want to ensure that the torso angle holds a constant pitch angle. We met all these objectives when the motor currents on the left, I_l, and right, I_r, are set equal to each other and to the constant pitch control given by Eqn. 1.

$$I_l = I_r = u_k \tag{2}$$

C. Turning motion

To turn, we add a differential current ΔI to the existing current on one motor and subtract an equal current on the other motor.

$$I_l = u_k + \Delta I, \qquad I_r = u_k - \Delta I, \qquad \text{Turn right;} \tag{3}$$

$$I_r = u_k + \Delta I, \qquad I_l = u_k - \Delta I, \qquad \text{Turn left.} \tag{4}$$

Since the motor current is proportional to output torque, torque is proportional to the pitch angle, pitch angle is proportional to the speed; we conclude that the motor current is proportional to the robot speed. Thus, the rimless wheel with more motor current will speedup relative to the other wheel, thus achieving turning. The rationale behind adding/subtracting the same current from the left and right rimless wheel is because the average current in the two motors determines the torso pitch angle, which is still u_k, and thus the torso pitch tries to achieve stabilization at r_k. We set the current differential ΔI manually using the joystick.

IV. Results

This section presents the experimental results. A video is in the reference [19]. Our metrics for the robot performance are the overall robot speed, the mean torque on the motors, the power drawn by the motors, the Total Cost Of Transport (TCOT).

$$\text{TCOT} = \frac{\text{Total Power}}{\text{Weight} \times \text{Velocity}} \tag{5}$$

A low TCOT implies a more energy-efficient motion.

A. Straight-line motion

To initialize the robot, we held the robot in a stationary position on two spokes, one on each side. We turn ON the torso pitch controller and allowed the torso to stabilize to a constant pitch angle. Once stabilized, we nudge the robot to impart slight momentum that helps it move forward. For a steady torso pitch angle, the robot achieves a set steady-state speed when initialized in this fashion.

The robot achieved its most energy-efficient movement on asphalt at a torso angle of 50°. The energy consumption was: raspberry PI and sensors 5 W, Teensy 0.2 W, and motors 3.6 W. The mean torque and speed were 1.41 Nm and 1.379 m/s respectively. Using the robot mass of 6.9 kg and Eqn. 5, we get $\text{TCOT} = 0.13$.

We tested the rimless wheel on different surfaces to study the effect of floor compliance on the TCOT and speed. Table I shows the results of testing on 5 different surfaces at steady state. For all these results we used a torso setpoint of 50°. As seen from the table, polished concrete has the lowest speed and highest TCOT while asphalt has the highest speed and lowest TCOT.

We also test the rimless wheel for different torso pitch angle. Theoretically, a torso pitch angle of 90° should give the fastest speed as it corresponds to maximum torque and hence the most energy added. However, because the torso pitch set-point control is not perfect, the torso would overshoot beyond 90° driving the system unstable. To offset the issue, we limit the torso pitch to a maximum of 85° to the vertically downward direction. The maximum speed achieved with this torso pitch angle is 4.32 m/s or 9.66 mph on polished wood, which was on an indoor basketball court.

B. Turning motion

To induce turning, we command the robot torso pitch to 50° and launched to move in a straight line. After the robot reaches a steady-state speed, it turns when commanding different currents in the motors using a hobby remote control.

TABLE I

Cost of Transport and velocities on different surfaces

Surface	TCOT	Avg. Velocity (m/s)
Polished Concrete	0.16	0.936
Polished Wood	0.15	1.197
Indoor Running Track	0.14	1.268
Outdoor Running Track	0.13	1.280
Asphalt	0.13	1.379

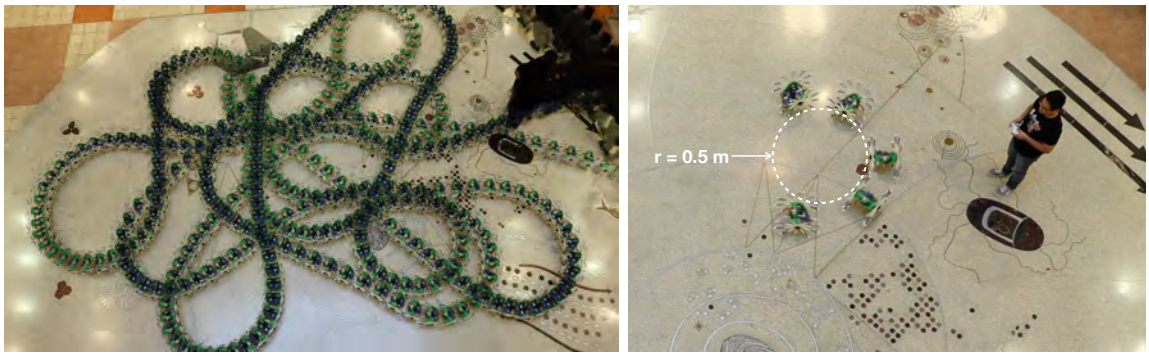

Fig. 3. **Experimental results for robot turning** (a) Trajectory of the robot during turning; (b) Shortest turning radius of 0.5 m.

Figure 3 (a) show the path taken by the robot. The robot can turn within the indoor facility without bumping into obstacles. The smallest turn radius achieved is 0.5 m as shown in Fig. 3 (b). We obtained the figure by super-posing 5 video frames and using a calibration scale to measure the radius. Here, we calculated the smallest turn radius by the circle traced by the innermost wheel in the turn. Note that the theoretical turn radius is zero (turning in place) corresponding to one wheel at zero speed and other turning at a non-zero speed. The issue is that the torque requirement for this is too low for the robot to hold a constant pitch angle of 50° leading to a loss of forward momentum, consequently coming to a halt. From multiple runs, we found that the robot turns more easily on hard floors at slow speeds, most likely because of the better traction between the legs and the floor at slow speeds.

V. Discussion

This paper presented the design and control of a differential drive rimless wheel robot that achieved a maximum speed of 9.66 mph and a minimum turn radius of 0.5 m. The 6.9 kg robot achieved a Total Cost of Transport (power per unit weight per unit speed) between 0.13 to 0.16 for different surfaces with speed ranging from 0.94 to 1.38 in m/s (2.1 to 3.1 in mph).

The Total Cost Of Transport (TCOT) defined as the power used per unit weight per unit velocity is the most widely accepted measure of energy usage of locomotion [20]. Humans have a TCOT of 0.3 while walking at a self-selected speed [21]. However, among 'true' bipedal robots (small feet relative to legs and only 2 legs), the most energy-efficient robots are Cornell biped with TCOT of 0.2 [22] and Cornell Ranger with a TCOT of 0.19 [23], [24]. The most energy-efficient legged robot is by ETH Zurich, Cargo at a TCOT of 0.1 [11] while RR 2 at TCOT of 0.13. However, Cargo has big feet and RR2 has multiple spokes. These features afford these robots enhanced stability with no additional controller considerations.

The RR2 with 20 legs in all reached a maximum speed of 4.32 m/s or 9.66 mph. The robot has legs of length 0.26 m which gives it a Froude number of 2.71 (Froude number is $v/\sqrt{g\ell}$ where v is speed, g is gravity 9.8 and ℓ is leg length). Outrunner, the 2 foot tall, 6 legged rimless wheel had a Froude number of 5.14 and HexRunner, the 6 foot tall, 6 legged rimless wheel had a Froude number of 4.82. Thus, RR2 is twice as slow as these other rimless wheels. To increase the speed of the rimless wheel, we need to reduce the collisional losses. We can do this either by decreasing its mass or increasing the number of spokes. Alternately one can add more energy to the system during the single stance phase by increasing the pitch angle or increasing the distance of the center of mass from the center of the wheel.

A differential drive-based turning is simple (only two motors needed) and enables a short turn radius. The minimum turning radius for the 0.31 wide RR2 is 0.5 m (see Fig. 3). The low net torque prevents us from achieving the shortest turn radius of 0 (turning in place) as discussed in Sec IV-B. The Outrunner and HexRunner use an oscillating pendulum-like mechanism in the lateral (side-ways) plane for turning, but the minimum turn radius has not been publicly available. The major difference between RR2 and these runners is that RR2 relies on kinematics to turn (speed of one wheel relative to the other), while the runners rely on the dynamics of the oscillating mass, the location of its center of mass and oscillating frequency. For the Outrunner/Hexrunner, ones needs to carefully tune the oscillation of the out-of-plane pendulum to ensure that the turning motion does not destabilize the robot in the side-ways direction.

The direct-drive brushless motor used in the robot has a KV rating of 280. This means that for a 1 V voltage output across the motor terminals, the input motor speed is 280 rpm. A high KV ensures high speed but a low KV ensures a high torque. This KV rating did not provide enough torque to support the weight of the torso. Thus we add a pulley transmission to increase the gearing by a factor of 5.4, thus increasing the torque without significantly affecting the speed.

A significant issue with the hobby-grade motors such as the ones used here is the torque ripple or the periodic fluctuation of the output torque for constant input current and load. This is because of the low number of magnetic cores in hobby-grade motors that lead to a lower magnetic strength as the rotor moves from one core to the next. The torque ripple causes controllability issues leading to a poor torque and/or position control. The torque ripple made it difficult to

achieve rapid modification in the speed and turning.

The rimless wheel had some failure modes. A successful launch was sensitive to the push. Too hard or too slow a push or asymmetrical push leads the rimless wheel to not launch properly, causing a failed trial. A fixture for standardizing the launch can solve this issue. Frequently, the rimless wheel ran into walls and broke one or more legs. However, because the legs were modular and 3D printed, it was relatively easy to fix the broken legs.

There are several limitations of the robot as discussed next. The robot cannot self-start as it needs to be launched manually. This is because there is an energy barrier to cross from the standing position with four spokes (2 on each side) touching the ground to the vertical upright position with two spokes touching the ground (1 on each side). One self-starting control strategy is pitching the torso back-and-forth appropriately to pump energy into the wheel. It is challenging to control the pumping with noisy sensors without leading to instability. Another self-starting means is to add an actuated tail or actuated spokes. The torque ripple prevents high fidelity motion control such as stabilizing against disturbance and quick turning. This issue may be circumvented by upgrading to motors with a higher number of magnetic cores [25] or by position or force-based feedback compensation [26]. The robot comes to stop quickly on rough terrain such as sandy and muddy terrain and is probably because of its relatively slow speed.

ACKNOWLEDGMENT

The authors would like to thank Ezra Ameperosa, Robert Brothers, Panchajanya Karasani, Emiliano Rodriguez, and Ali Zamani for help with testing. The work was partially supported by NSF grant IIS 1946282 and 2010736 to PAB.

REFERENCES

[1] T. McGeer, "Passive dynamic walking," *The International Journal of Robotics Research*, vol. 9, no. 2, p. 62, 1990.

[2] P. A. Bhounsule, "Numerical accuracy of two benchmark models of walking: the rimless spoked wheel and the simplest walker," *Dynamics of Continuous, Discrete and Impulsive Systems, Series B: Applications and Algorithms*, p. 137—148, 2014.

[3] M. J. Coleman, A. Chatterjee, and A. Ruina, "Motions of a rimless spoked wheel: a simple three-dimensional system with impacts," *Dynamics and stability of systems*, vol. 12, no. 3, pp. 139–159, 1997.

[4] A. C. Smith and M. D. Berkemeier, "The motion of a finite-width rimless wheel in 3d," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, vol. 3. IEEE, 1998, pp. 2345–2350.

[5] M. W. Gomes and K. Ahlin, "Quiet (nearly collisionless) robotic walking," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5761–5766.

[6] S. K. Agrawal and J. Yan, "A three-wheel vehicle with expanding wheels: Differential flatness, trajectory planning, and control," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 2. IEEE, 2003, pp. 1450–1455.

[7] D. Laney and D. Hong, "Kinematic analysis of a novel rimless wheel with independently actuated spokes," in *29th ASME Mechanisms and Robotics Conference, Long Beach, California*, 2005.

[8] S. Cotton, J. C. Godowski, N. R. Payton, M. Vignati, C. Schmidt-Wetekam, C. Black *et al.*, "Multi-legged running robot," Jan. 7 2016, uS Patent App. 14/596,514.

[9] W. Isern, "Ihmc robot breaks speed record," https://www.pnj.com/story/news/2014/06/08/ihmc-robot-breaks-speed-record/10211537/, March 2019 2019.

[10] P. A. Bhounsule, E. Ameperosa, S. Miller, K. Seay, and R. Ulep, "Dead-beat control of walking for a torso-actuated rimless wheel using an event-based, discrete, linear controller," in *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2016, pp. V05AT07A042–V05AT07A042.

[11] F. Guenther and F. Iida, "Energy-efficient monopod running with a large payload based on open-loop parallel elastic actuation," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 102–113, 2016.

[12] W. Sirichotiyakul, A. C. Satici, S. Sanchez, and P. A. Bhounsule, "Energetically-optimal discrete and continuous stabilization of the rimless wheel with torso," in *ASME-International Design Engineering & Technical Conference*, 2019.

[13] P. A. Bhounsule, "All files related to roadrunner2, a differential drive rimless wheel," https://github.com/pab47/RoadRunner2, May 2020.

[14] O. robotics, "Dual shaft motor d5065 270kv," https://odriverobotics.com/shop/odrive-custom-motor-d5065, March 2019.

[15] R. P. Foundation, "Raspberry pi 3 model b," https://www.raspberrypi.org/products/raspberry-pi-3-model-b/, March 2019.

[16] PJRC, "Teensy usb development board," https://www.pjrc.com/store/teensy32.html, March 2019.

[17] O. robotics, "Odrive high performance motor control," https://odriverobotics.com/shop/odrive-v35, March 2019.

[18] T. Wescott, "Pid without a phd," *Embedded Systems Programming*, vol. 13, no. 11, pp. 1–7, 2000.

[19] P. A. Bhounsule, "A differential drive rimless wheel," https://youtu.be/SNxeP29ayhI, April 2020.

[20] T. Von Karman and G. Gabrielli, "What price speed? specific power required for propulsion of vehicles," *Mechanical Engineering*, vol. 72, pp. 775–781, 1950.

[21] A. Bobbert, "Energy expenditure in level and grade walking," *Journal of Applied Physiology*, vol. 15, no. 6, pp. 1015–1021, 1960.

[22] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, p. 1082, 2005.

[23] P. Bhounsule, A. Ruina *et al.*, "Cornell ranger: energy-optimal control," *Dynamic walking*, 2009.

[24] P. A. Bhounsule, J. Cortell, A. Grewal, B. Hendriksen, J. D. Karssen, C. Paul, and A. Ruina, "Low-bandwidth reflex-based control for lower power walking: 65 km on a single battery charge," *The International Journal of Robotics Research*, vol. 33, no. 10, pp. 1305–1321, 2014.

[25] W. Qian, S. K. Panda, and J.-X. Xu, "Torque ripple minimization in pm synchronous motors using iterative learning control," *IEEE Transactions on Power Electronics*, vol. 19, no. 2, pp. 272–279, 2004.

[26] M. Piccoli and M. Yim, "Anticogging: Torque ripple suppression, modeling, and parameter selection," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 148–160, 2016.