

DETC2016-59563

DEAD-BEAT CONTROL OF WALKING FOR A TORSO-ACTUATED RIMLESS WHEEL USING AN EVENT-BASED, DISCRETE, LINEAR CONTROLLER

Pranav A. Bhounsule, Ezra Ameperosa, Scott Miller, Kyle Seay, Rico Ulep

Robotics and Motion Laboratory
Dept. of Mechanical Engineering,
University of Texas San Antonio

One UTSA Circle, San Antonio, TX 78249, USA.

Corresponding author email: pranav.bhounsule@utsa.edu

ABSTRACT

In this paper, we present dead-beat control of a torso-actuated rimless wheel model. We compute the steady state walking gait using a Poincaré map. When disturbed, this walking gait takes a few steps to cancel the effect of the disturbance but our goal is to develop a faster response. To do this, we develop an event-based, linear, discrete controller designed to cancel the effect of the disturbance in a single step – a one-step dead-beat controller. The controller uses the measured deviation of the stance leg velocity at mid-stance to set the torso angle to get the wheel back to the limit cycle at the following step. We show that this linear controller can correct for a height disturbance up to 3% leg length. The same controller can be used to transition from one walking speed to another in a single step. We make the model-based controller insensitive to modeling errors by adding a small integral term allowing the robot to walk blindly on a 7° uphill incline and tolerate a 30% added mass. Finally, we report preliminary progress on a hardware prototype based on the model.

1 Introduction

Gravity powered legged robots use gravitational potential energy for locomotion. The most famous examples are the passive dynamic robots developed by McGeer [1]. Passive dynamic legged robots consists of links joined together by hinges. When these robots are launched downhill with the right set of initial conditions, they are able to sustain steady locomotion. These

robots are highly energy-effective because they use no motors or external power and have a zero energy cost.

There are two major limitation of passive, downhill descending legged robots pioneered by McGeer: (1) they cannot sustain walking on level ground and uphill (2) they are not robust to external disturbances. In this research, we present a rimless wheel robot which is gravity powered by virtue of its leaning torso. We develop a control strategy that uses velocity regulation between steps to enable the following: (1) robust rough terrain locomotion (height changes, slopes); (2) variable speed locomotion (ability to change speeds quickly); and (3) ability to carry added mass. Finally, we show preliminary work towards developing a hardware prototype based on the model and analysis.

2 Background and Related Work

The passive rimless wheel robot consists of a spoked wheel without a rim descending downhill [1]. When launched on a ramp, the rimless wheel has one stable walking solution [2]. The rimless wheel loses energy at every step due the collision of the spoke with the ground. For the downhill walking rimless wheel, this energy is made up by gravity leading to sustained downhill walking. But the simplest rimless wheel cannot walk on level ground unless actuated in some way.

One way to enable sustained locomotion of a rimless wheel on a level ground is to have a design that ensures that the collisional losses are reduced to zero. This can be done by ensuring

that the velocity of the spoke at touchdown is zero [3]. Gomes and Ahlin [4] designed a mechanical contraption in which the rimless wheel is coupled to an inertial disc through a torsional spring. The role of the torsional spring is to decelerate the rimless wheel just as it is about to have a collision and transfer the energy to the inertia disc so that the spoke makes a nearly zero velocity collision with the ground. During the first half of the step, the torsional spring provides the energy from the inertia disc to the rimless wheel. Such a robot can sustain steady state motions on level ground in simulation but will need a shallow ramp in reality because of the dissipation in the springs, inertia disc, and friction losses in the bearings.

Another way to enable level ground walking is to have an actively powered rimless wheel. There are multiple ways of doing this. One idea is to have spokes which expand and contract supplying energy to the robot [5]. However, this requires multiple motors. Another idea is to have a wobbling mass [6] or a rotating disc [7] that transfers energy by dynamic coupling. Our design is slightly different. We add a torso to the rimless wheel which provides gravitational energy by leaning forward and is practically the same as the Outrunner [8]. For an exhaustive stability analysis of passive and controlled rimless wheel robots see Asano [9].

Dead-beat control refers to full correction of the deviations of the system in finite time [10]. In the case of legged robots, this would be the ability to correct the deviation in finite steps. Dead-beat control is attractive because it allows the robot to fully recover from external disturbances. The control scheme is relatively straightforward. A steady state gait is expressed as a limit cycle (see Sec. 4.2 for details). The deviations from the limit cycle can be corrected by controlling the foot placement [11, 12] or some other gait parameter such as leg stiffness [13]. The number of steps needed for full correction depends on the number of variables to be regulated and the number of control parameters. If there are n goals and n control actions, then it would take a single step to correct for the deviation assuming that none of the control parameters are saturated [14]. In this paper, we present an event-based, linear, discrete controller that can correct for disturbances in a single step.

The main novelty of this work is development of a simple event-based, linear controller that cancels the effect of the disturbance in a single step. The same controller can be used for one step velocity regulation too. The controller is simple because it uses a single measurement (stance leg velocity at mid-stance) and sets one control variable (torso angle) to correct for disturbances in a single step. Simple extension to the controller, like the addition of an integral term, can make the controller insensitive to modeling effects and unknown terrain variation.

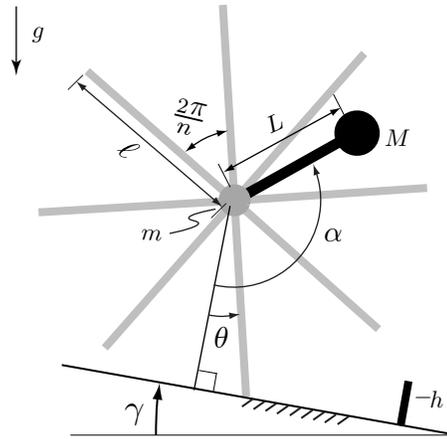


FIGURE 1. MODEL OF THE RIMLESS WHEEL ROBOT WITH TORSO. THE ROBOT IS MOVING TO THE RIGHT. THERE IS AN ACTUATOR BETWEEN THE TORSO AND THE HUB THAT CAN CONTROL THE TORSO ANGLE, α .

3 Model and Numerical Simulation

The model of the robot is shown in Fig. 1. The rimless wheel consists of $N = 8$ massless spokes, each of length ℓ . The mass of the rimless wheel is m and is concentrated at the hub. A point mass M is attached to the hub through a massless rod of length L and constitutes the torso. There are two phases of motion. The single stance phase where the entire rimless wheel pivots about a single spoke, moving like an inverted pendulum (see Sec. 3.1.1) and the heel-strike phase where there is an exchange of support from one spoke to another through a plastic collision (see Sec. 3.1.2). The condition for transition from single stance to heel-strike is given in Sec. 3.1.3. The gravity is g and points downwards. The ramp slope, γ , and height, h , are used to create a disturbance, but are normally set to zero. The contact spoke and the torso make an angle of θ and α respectively with the normal to the ramp.

The only control variable for this model is the torso angle. To make analysis simple we assume that the torso angle is set to α instantaneously once per step. Note that α is the angle with respect to the inertial frame, so the actuator needs to continuously control the torso during the step to maintain this orientation. On hardware this can be easily done by using an inertial measurement unit that measures the orientation with respect to the inertial frame. Also, we neglect the transient response when the actuator sets the torso angle during the step. This is a fair assumption to make because the actuator, when tuned properly, can quickly change the torso angle without creating unnecessary transients.

3.1 Equations of motion

We are going to give very brief explanation about deriving the equations of motion. Please refer to the thesis [15] for a more detailed derivation of the equations.

3.1.1 Equations for single stance phase: The equation of motion for the single stance phase can be obtained by doing an angular momentum balance about the spoke which contacts the ground to obtain the following ordinary differential equation.

$$\begin{aligned} A\ddot{\theta} &= b_{ss}, \\ A &= \{m\ell^2 + M\ell(\ell - L\cos(\theta - \alpha))\}, \\ b_{ss} &= M\ell L\dot{\theta}^2 \sin(\alpha - \theta) + mg\ell \sin(\theta - \gamma) + \\ &Mg\{\ell \sin(\theta - \gamma) - L\sin(\alpha - \gamma)\}. \end{aligned} \quad (1)$$

We have included a downhill slope γ (with respect to the horizontal) in the equations. This is used to add a disturbance to the model to check the robustness of the controller.

3.1.2 Equations for heel-strike phase: The equation of motion for the heel-strike is obtained by applying conservation of angular momentum about the point where the following foot is about to hit and is given by an algebraic equation

$$\theta^+ = \arccos\left(\cos(\theta^-) + \frac{h}{\ell}\right) \quad (2)$$

$$A\dot{\theta}^+ = b_{hs} \quad (3)$$

where A is the same as that in Eqn. 1 and

$$b_{hs} = \left((M+m)\ell^2 \cos\left(\frac{2\pi}{N}\right) - M\ell L \cos(\alpha - \theta^-)\right)\dot{\theta}^-$$

3.1.3 Equations for detecting a collision: The condition for transition from single stance to heel-strike is.

$$\ell \cos(\theta) - \ell \cos\left(\theta + \frac{2\pi}{N}\right) + h = 0. \quad (4)$$

We have included step down, h , in the equations. This is used to add a disturbance to the model to check robustness of the controller.

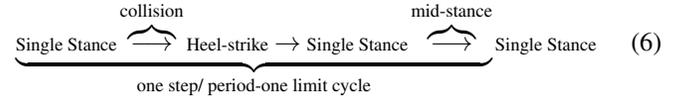
3.1.4 Equations for detecting mid-stance: The mid-stance is the position when the stance leg is vertical with respect to the slope, and is given by the following condition

$$\theta = 0. \quad (5)$$

This position is used by the controller to adjust the torso angle for control. More details are in the next section.

3.2 Simulation

3.2.1 A single step At the core of the simulator is a single step that starts from mid-stance and ends at the mid-stance of the next step. The initial state for the robot is the mid-stance position, $\{0, \dot{\theta}^m\}$, where $\dot{\theta}^m$ is the stance leg velocity at mid-stance. Given the initial state at mid-stance, the robot goes through the following sequence of motions



In the above sequence single stance is given by Eqn. 1, the collision event is given by Eqn. 4, the heel-strike is given by Eqns. 2 and 3, and mid-stance event is given by Eqn. 5. The equations for single stance are integrated in MATLAB using *ode113*¹ with an error tolerance of 10^{-9} . The two events are detected by the event detector in the integrator.

3.2.2 Failure modes There are two failure modes for the robot which we need to check at each integration step. These are described next.

Falling Backwards: Falling backwards is detected by checking the angle made by the spoke with the vertical. This happens when the angle between the stance leg angle and gravity is more than half the inter-spoke angle. The condition is $\theta > \frac{\pi}{N}$.

Flight phase: When the vertical ground reaction force, R_y , on the stance leg goes to zero, we get a flight phase. Since we are restricted to walking motions, we consider this to be a failure mode. This condition is

$$R_y = (M+m)(g - \dot{\theta}^2 \cos(\gamma - \theta) + \ddot{\theta} \sin(\gamma - \theta)) = 0.$$

4 Methods

4.1 Poincaré Map:

The Poincaré map is used to analyze walking [1, 16]. To compute the map, we need to relate the state of the walker at any instance in the step with the same instance on the next step. Here, we will relate the state at mid-stance on the current step, n , to the mid-stance at the next step, $n+1$. To do this, we can use Eqns. 1 to 5 to get

$$\dot{\theta}_{n+1}^m = F(\dot{\theta}_n^m, \alpha_n, \gamma_n, h_n), \quad (7)$$

¹We have found MATLAB ode integrator, *ode113*, to be slightly faster for stringent tolerances than the more commonly used function, *ode45*.

where the Poincaré mapping, F , is a scalar function that maps the mid-stance angular velocity at step n , $\dot{\theta}_n^m$, to the mid-stance angular velocity at step $n+1$, $\dot{\theta}_{n+1}^m$. The torso angle, α_n , is a control parameter that we can adjust as desired. Finally, the slope, γ_n , and the height drop at heel-strike, h_n , is an external disturbance that we can set. We numerically solve the equations (see Sec. 5).

4.2 Period-one limit cycle for steady state walking:

Limit cycles are periodic solutions of the Poincaré map F . To find the limit cycle, we need to find a fixed point of the function F . We compute limit cycles for level ground by putting $\gamma_n = \gamma = 0$. Also, since there is no height drop at heel-strike, thus $h_n = h = 0$. We put these values in Eqn. 7 and put steady state value for velocity, $\dot{\theta}_{n+1}^m = \dot{\theta}_n^m = \dot{\theta}^m$. Also, this steady state speed will correspond to a certain torso angle, $\alpha_n = \bar{\alpha}$, say.

We can rewrite Eqn. 7 for a limit cycle as follows:

$$\dot{\theta}^m = F(\dot{\theta}^m, \bar{\alpha}, \gamma = 0, h = 0). \quad (8)$$

The torso angle, $\bar{\alpha}$, is a control variable. Hence the only unknown is the mid-stance speed, $\dot{\theta}^m$. Thus the limit cycle is characterized by a single variable, namely, the angular speed at mid-stance $\dot{\theta}^m$.

4.3 Eigenvalue based stability

For limit cycle walkers, stability is associated with the ability of the robot to be on the same limit cycle in the presence of a disturbance (e.g., terrain variation, a push). The stability is found by finding the Jacobian of the limit cycle (F in Eqn. 8) and then finding the eigenvalues of the Jacobian. This is given by

$$J = \frac{\partial F}{\partial x}, \quad (9)$$

$$\lambda = \text{eig}(J), \quad (10)$$

where the state space is $x = \{\theta, \dot{\theta}\}$ and the two eigenvalues are in λ . Because we are computing the eigenvalues of a Poincaré map, one eigenvalue is zero and corresponds to a perturbation in the absolute position of the stance leg. Thus, the system has only one non-trivial eigenvalue. The walking machine is stable if the magnitude of the eigenvalue is less than 1 and unstable if the eigenvalue is greater than 1 [17].

4.4 Discrete, once per step, linear control

If the biggest eigenvalue is λ_{\max} , then the effect of disturbance will grow/shrink by a factor of λ_{\max} at every step. For example, with an eigenvalue is 0.56, it takes about 4 steps for

the deviation to reduce by a factor of 100 ($0.56^4 \approx 0.01$). In this case, one might want to design a controller that can get the system back to the limit cycle at a much faster rate. In this section, we indicate how to develop a linear, discrete, event-based controller that gets the rimless wheel back to the limit cycle in a single step, also known as a one-step dead-beat controller [10].

We first linearize Eqn. 8 at the fixed point to get

$$\Delta \dot{\theta}_{n+1}^m = A_{\dot{\theta}} \Delta \dot{\theta}_n^m + B_{\alpha} \Delta \alpha. \quad (11)$$

Next we assume a linear controller, $\Delta \alpha = -K \Delta \dot{\theta}_n^m$ and put it in the above equation to get

$$\Delta \dot{\theta}_{n+1}^m = (A_{\dot{\theta}} - B_{\alpha} K) \Delta \dot{\theta}_n^m \quad (12)$$

To develop a one-step dead beat controller we set $\Delta \dot{\theta}_{n+1}^m = 0$ and solve for K

$$K = B_{\alpha}^{-1} A_{\dot{\theta}} \implies \Delta \alpha = -B_{\alpha}^{-1} A_{\dot{\theta}} \Delta \dot{\theta}_n^m. \quad (13)$$

The gain K is a scalar and inverting B_{α} is not an issue. If B_{α} is not square, then one can use the pseudo-inverse. Alternately, in MATLAB, K can be obtained by using the *place* command. However, one restriction is that $\alpha < \pi$ for the rimless wheel to keep moving forward. The controller that we describe above is event-based because it is triggered at mid-stance, is discrete because of Eqn. 11, and is linear because of Eqns. 12 and 13. By setting the eigenvalue to zero in Eqn. 13, we get a one-step dead-beat controller. More details about this type of controller is in [18, 19].

5 Numerical Results

5.1 Model parameters

The model parameters for simulations are shown in Table 1. The torso angle α is a control variable that we set once per step. The nominal walking is on level ground, so the ramp slope, i.e., $\gamma = 0$ (Eqn. 1), and there is no step variation during heel-strike, i.e., $h = 0$ (Eqn. 4).

5.2 Limit cycle

To find the limit cycle, we need to solve Eqn. 8. To do this, we use the non-linear equation solver, *fsolve* in MATLAB. We specify a tolerance of 10^{-6} for *fsolve*. A torso angle, $\alpha = 0.5401$, gives an average speed of 1.33 m/s, which corresponds to average human walking speed. We define average speed as the ratio of step length to step time.

The fixed point is $x^* = \{0, -1.2057\}$. Figure 2 shows the position and velocity for steady state walking of the rimless wheel.

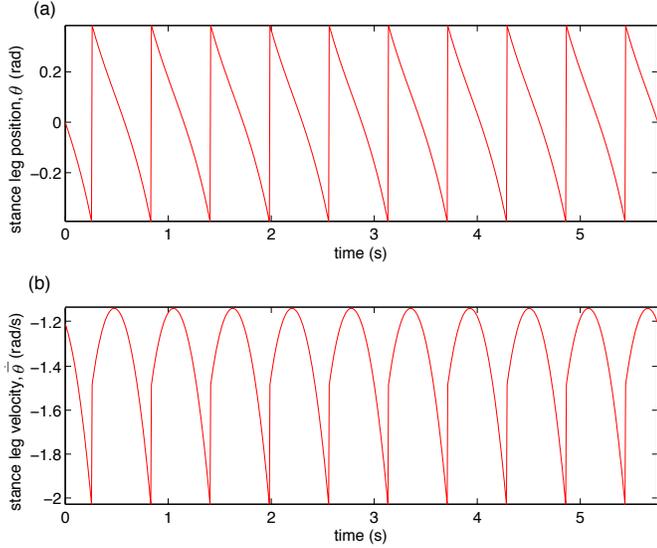


FIGURE 2. PERIOD ONE LIMIT CYCLE FOR THE RIMLESS WHEEL: (A) STANCE LEG POSITION AS A FUNCTION OF TIME AND (B) STANCE LEG VELOCITY AS A FUNCTION OF TIME. THE SPEED OF THE RIMLESS WHEEL IS 1.33 m/s .

TABLE 1. MODEL PARAMETERS FOR SIMULATIONS

Parameter	Value
N	8
$m = M$	1 kg
$\ell = 2L$	1 m
g	9.81 m/s^2

5.3 Eigenvalues

We compute the Jacobian of the Poincaré map (Eqn. 9) by doing a central difference of size 10^{-3} . Next, we compute the eigenvalues by using the function *eig* in MATLAB to get, $\lambda = \{0, 0.4301\}$. The first eigenvalue is zero and corresponds to a perturbation in the position direction. Since our map is from mid-stance to mid-stance, the perturbation in the position direction is on the Poincaré section and hence evaluates to zero. Thus, the maximum eigenvalue is, $\lambda_{\max} = 0.4301 < 1$ and indicates a stable limit cycle. However, it would take the walker about 6 steps for a perturbation to reduce by a factor of 100. That is, $0.4301^6 \approx 0.01 \approx 0$. In the next section we will develop a controller that can get the robot back to the limit cycle in a single step.

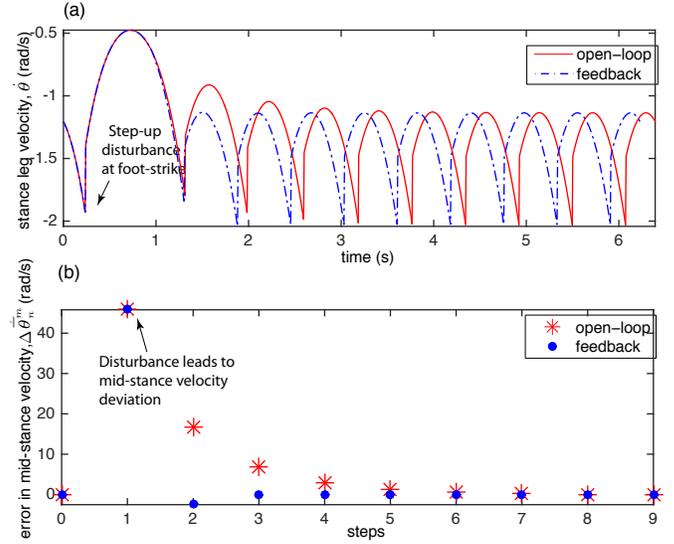


FIGURE 3. DISTURBANCE REJECTION OF THE CONTROLLER. A STEP DOWN OF 0.03 LEG LENGTH IS INTRODUCED AT THE FIRST HEEL-STRIKE, APPROX 0.1 SEC IN (A). THIS LEADS TO A 45% DECREASE IN VELOCITY AT THE NEXT MID-STANCE IS SHOWN IN (B). WITHOUT A CONTROLLER (OPEN-LOOP) IT TAKES ABOUT 6 STEPS TO GET TO THE LIMIT CYCLE (RED STARS) BUT IT TAKES TWO STEPS FOR OUR FEEDBACK CONTROLLER TO CORRECT FOR DISTURBANCE (BLUE DOTS).

5.4 Linearized equations for control

The linearized control equation (Eqn. 11) is obtained by using a central difference step size of 10^{-3} . We get

$$\Delta \hat{\theta}_{n+1}^m = 0.4301 \Delta \hat{\theta}_n^m - 1.4789 \Delta \alpha \quad (14)$$

Note that the first variable $A_{\hat{\theta}} = 0.4301$ is the eigenvalue of the system. The value of $B_{\alpha} = -1.4789$ is negative. This suggests that a positive increment in α will make the mid-stance velocity at the next step, $\hat{\theta}_{n+1}^m$, to increase and become more negative. This makes sense because by increasing the torso angle, the gravity potential increases leading to a faster speed.

We obtain one-step dead-beat controller by substituting values for $A_{\hat{\theta}}$ and B_{α} into Eqn. 13 to get

$$K = -0.2908 \Rightarrow \Delta \alpha = 0.2908 \Delta \hat{\theta}_n^m. \quad (15)$$

5.5 Disturbance rejection

Disturbance rejection is the ability of the robot to reject exogenous disturbances. Current legged robots are most fragile to

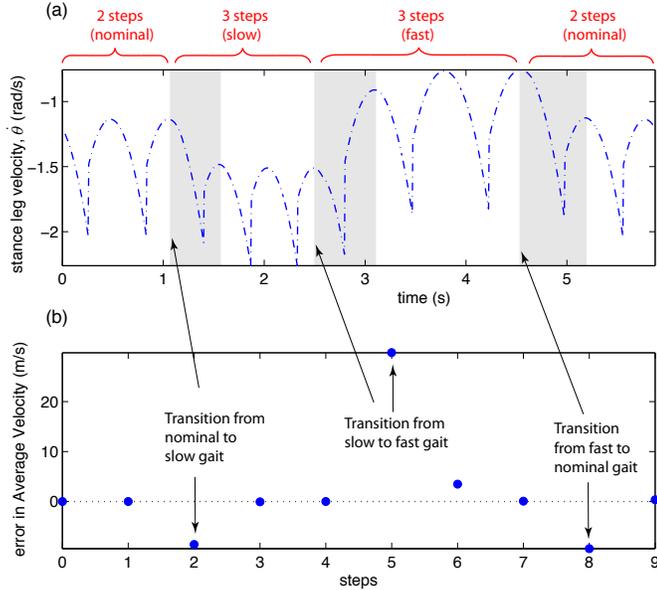


FIGURE 4. VARIABLE SPEED WALKING: 2 STEPS AT THE NOMINAL GAIT, 3 STEPS AT THE SLOW GAIT, 3 STEPS AT THE FAST GAIT, 2 STEPS AT THE NOMINAL GAIT. THE ONE-STEP DEAD-BEAT CONTROLLER SWITCHES BETWEEN GAITS IN A SINGLE STEP.

terrain variation [20]. To test our feedback controller, we introduce a step up of 0.03ℓ at the end of step 0 as shown in Figure 3 (a). The effect is that the mid-stance velocity decreases by 45 % at mid-stance at step 1. Beyond a step up (height) disturbance of 0.03ℓ , the robot falls backwards which is a failure mode. With open loop control, the rimless wheel takes about 6 steps to get to the nominal speed, while with our feedback controller (Eqn. 15), the robot gets to nominal speed on the two-steps. This suggests that the linearization of the step height disturbance to the robot mid-stance velocity is non-linear, else we would have obtained a one-step dead-beat response. Similarly, the controller is able to correct for a step down (height) disturbance of 0.03ℓ . Any bigger step down leads to a flight phase which we consider as a failure mode and terminate the simulation.

5.6 Variable speed walking

There might be instances when we want the rimless to walk at different speeds (e.g., to meet certain overall velocity constraint). This will require the rimless wheel to switch between speeds quickly. The one-step dead-beat controller can be used to do this transition. We illustrate this next.

We have computed open-loop and feedback gains for three walking speeds as shown in Table 2. These are for walking at speeds to 1 m/s (slow gait), 1.33 m/s (nominal gait), and 1.66

TABLE 2. THREE WALKING SPEEDS AND ASSOCIATED CONTROLLER SENSITIVITIES AND FEEDBACK GAIN.

Parameter	Units	Nominal	Fast	Slow
V	m/s	1.33	1.66	1.00
α	rad	0.5401	0.712	0.421
$\dot{\theta}^m$	rad/s	-1.2057	-1.5763	-0.8325
$A_{\dot{\theta}}$	-	0.4301	0.4489	0.4178
B_{α}	1/s	-1.4789	-0.9949	-2.2707
K	s	-0.2908	-0.4512	-0.1840

m/s (fast gait). We want the following walking sequence.

$$\underbrace{\text{Nominal Gait}}_{2 \text{ steps}} \Rightarrow \underbrace{\text{Slow Gait}}_{3 \text{ steps}} \Rightarrow \underbrace{\text{Fast Gait}}_{3 \text{ steps}} \Rightarrow \underbrace{\text{Nominal Gait}}_{2 \text{ steps}} \quad (16)$$

It is straightforward to realize this walking sequence by switching between controllers at mid-stance after the required number of steps. Figure 4 (a) shows the stance leg velocity vs time and (b) shows the error in mid-stance speed as a function of step length. During the transition, the error in mid-stance speed is not zero (see Fig. 4 (b) steps 2, 5, and 8). This error in speed is nulled by the feedback controller in a single step. Thus, switching gaits is quite straightforward with our control approach.

5.7 Integral control for handling modelling errors

Our linear, discrete, event-based controller is a model-based controller that cannot handle modeling errors. We consider two sources of errors; a steady slope and an added mass on the hub. For these two errors, our controller will give a steady state error. However, if our controller is supplemented with an integral term over the mid-stance speed deviation, the resulting controller can be made insensitive to modeling errors. Thus the controller takes the following form

$$\Delta \alpha = -K \Delta \dot{\theta}_n^m + K_I \int \Delta \dot{\theta}_n^m, \quad (17)$$

where K_I is the integral control gain and has the effect of reducing the steady state value to zero. The gain is kept relatively small so that it does not affect the one-step dead-beat control presented earlier.

We consider the effect of slope and added mass on the nominal gait which corresponds to a speed of 1.33 m/s . We set a

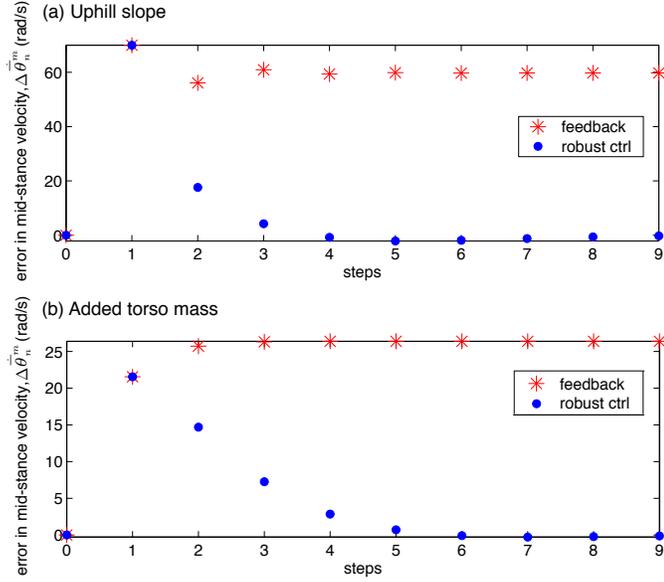


FIGURE 5. (A) ROBUSTNESS AGAINST TERRAIN VARIATION. THERE IS A 7° UPHILL. (B) ROBUSTNESS AGAINST ADDED MASS. A MASS OF 1 KG IS ADDED TO THE HUB WHICH CONSTITUTES A 30% INCREASE IN MASS. IN EITHER CASE, THE MODEL-BASED FEEDBACK CONTROLLER (RED ASTERISCS) SHOWS A STEADY STATE ERROR WHILE THE FEEDBACK CONTROLLER WITH ADDED INTEGRAL TERM IS ABLE TO REGULATE TO THE STEADY-STATE SPEED (BLUE CIRCLES)

slope, $\gamma = -0.1221 = -7^\circ$, where a negative value indicate an uphill slope. We simulate the system with the event-based controller with and without integral term. Only the controller with the integral term is able to maintain the desired nominal speed. This is shown in Fig. 5 (a). Similarly, we add a mass of 1 kg at the hub and simulate the system with and without the integral term. The integral controller was able to nullify the effect of the added mass in a couple of steps and is shown in Fig. 5 (b).

6 Hardware

We discuss preliminary results on developing a hardware prototype based on the rimless wheel robot. Our robot is called the Roadrunner and is named after the mascot of the University and is shown in Fig.6 (a).

The robot consists of two rimless wheel sub-assemblies attached to each other through an axle which holds a torso (the box) that houses the motors, batteries, and electronics. The robot legs and the torso are 3D printed on Stratasys Dimension 1220es 3-D printer. The axle is of aluminum and there are compression springs in each leg to cushion the collision. The Fig.6 (b) shows a schematic of the motors, power train, batteries, and elec-

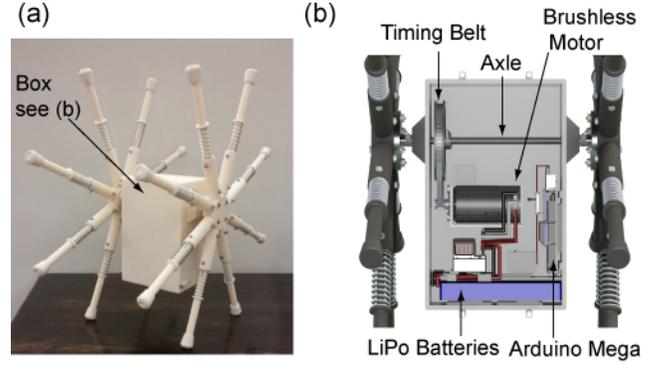


FIGURE 6. (A) THE "ROADRUNNER" RIMLESS WHEEL ROBOT (B) SCHEMATIC OF THE TORSO (BOX) BETWEEN THE RIMLESS WHEEL SUBASSEMBLIES THAT HOUSES THE MOTORS, THE BATTERIES AND THE ELECTRONICS.

tronics. The 22.2 V brushless DC motor connects to the axle by a timing belt. The motor is powered by two 3S Lithium-ion Polymer batteries connected in series. An Arduino Mega 2560 is used to control the motor. A 6-axis inertial measurement from Sparkfun is also placed in the torso. The gyroscope values and accelerometer values are fused together using a balance filter to compute the orientation of the torso with respect to the vertical. A Proportional-Integral Controller on the torso orientation is used to control the torso angle.

In preliminary trials, we got the Roadrunner to move at a speed of 6 miles per hour (approximately 2.68 m/s). Due to lack of data collection capabilities in our current robot setup, we are not able to provide experimental results, except for a video that is in reference [21]. Note how the torso leans forwards as the robot begins to accelerate. The motion of our robot is very similar to that of the Outrunner robot [8, 22] but the exact mechanism used for forward motion for Outrunner is not very clear to us.

The major difference between the model and the rimless wheel prototype is that the prototype has springy legs while the model does not. To develop a controller for the hardware prototype, Eqn. 11 will have to be modified as follows:

$$\Delta \dot{\theta}_{n+1}^m = \bar{A}_\theta \Delta \dot{\theta}_n^m + \bar{B}_\alpha \Delta \alpha \quad (18)$$

where \bar{A}_θ and \bar{B}_θ are appropriate sensitivities. One can derive an expression for the sensitivities using a rimless wheel model with spring. Another method would be to evaluate the sensitivities directly on the robot by perturbing the mid-stance velocity and the torso angle. Currently, we do not have a sensor to measure the robot speed but we are modifying the robot to incorporate an encoder. We plan to test the control approach on the robot in near future.

7 Conclusions and Future Work

In this paper, we presented the simplest gravity powered model of rimless wheel capable of walking on level ground in addition to slopes. The modification is the addition of an actuated torso that can be set to a prescribed set point instantaneously. The actuated torso is analogous to the ramp of a passive rimless wheel. By controlling the torso angle, the rimless wheel can generate walking gaits at various speeds. We developed event-based, discrete, linear controller capable of rejecting disturbances in a single step. The same controller can be used to switch gaits by interpreting the deviation of one controller as a disturbance for the transitioning gait. By adding an integral controller the controller can be made insensitive to modeling disturbances.

Our future work will focus on extended trials on the rimless wheel prototype to realize robust walking gaits. We have neglected running gaits but our framework can easily extend to running motions as well. Finally, the model can be extended to 3D by adding a finite width [23, 24] and adding another actuator in the side-to-side direction for the torso to allow for turning.

ACKNOWLEDGMENT

The authors would like to thank Hung-da Wan for access to Stratasys 3D printer and to John Simonis and James Johnson for mentoring and guidance on the hardware prototype.

REFERENCES

- [1] McGeer, T., 1990. "Passive dynamic walking". *The International Journal of Robotics Research*, **9**(2), pp. 62–82.
- [2] Coleman, M. J., 2010. "Dynamics and stability of a rimless spoked wheel: a simple 2d system with impacts". *Dynamical Systems*, **25**(2), pp. 215–238.
- [3] Gomes, M., and Ruina, A., 2011. "Walking model with no energy cost". *Physical Review E*, **83**(3), p. 032901.
- [4] Gomes, M. W., 2005. "Collisionless rigid body locomotion models and physically based homotopy methods for finding periodic motions in high degree of freedom models". PhD thesis, Cornell University.
- [5] Jeans, J. B., and Hong, D., 2009. "Impass: Intelligent mobility platform with active spoke system". In Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE, pp. 1605–1606.
- [6] Asano, F., Sogawa, T., Tamura, K., and Akutsu, Y., 2013. "Passive dynamic walking of rimless wheel with 2-dof wobbling mass". In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, IEEE, pp. 3120–3125.
- [7] Asano, F., and Xiao, X., 2012. "Output deadbeat control approaches to fast convergent gait generation of underactuated spoked walker". In System Integration (SII), 2012 IEEE/SICE International Symposium on, IEEE, pp. 265–270.
- [8] Robotics Unlimited, 2015. Meet outrunner: The world's first remotely controlled running robot. Youtube link https://youtu.be/LTIpdtv_AK8.
- [9] Asano, F., 2015. "Fully analytical solution to discrete behavior of hybrid zero dynamics in limit cycle walking with constraint on impact posture". *Multibody System Dynamics*, pp. 1–23.
- [10] Antsaklis, P., and Michel, A., 2006. *Linear systems*. Birkhauser.
- [11] Peucker, F., Maufroy, C., and Seyfarth, A., 2012. "Leg-adjustment strategies for stable running in three dimensions". *Bioinspiration & biomimetics*, **7**(3), p. 036002.
- [12] Wu, A., and Geyer, H., 2013. "The 3-d spring-mass model reveals a time-based deadbeat control for highly robust running and steering in uncertain environments". *Robotics, IEEE Transactions on*, **29**(5), pp. 1114–1124.
- [13] Saranlı, U., Schwind, W. J., and Koditschek, D. E., 1998. "Toward the control of a multi-jointed, monopod runner". In Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on, Vol. 3, IEEE, pp. 2676–2682.
- [14] Carver, S., Cowan, N., and Guckenheimer, J., 2009. "Lateral stability of the spring-mass hopper suggests a two-step control strategy for running". *Chaos: An Interdisciplinary Journal of Nonlinear Science*, **19**, pp. 026106–14.
- [15] Coleman, M., 1998. "A stability study of a three-dimensional passive-dynamic model of human gait". PhD thesis, Cornell University.
- [16] Garcia, M., Chatterjee, A., Ruina, A., and Coleman, M., 1998. "The simplest walking model: Stability, complexity, and scaling". *ASME J. of Biomech. Eng.*, **120**, pp. 281–288.
- [17] Strogatz, S., 1994. *Nonlinear dynamics and chaos*. Addison-Wesley Reading.
- [18] Bhounsule, P. A., Ruina, A., and Stiesberg, G., 2015. "Discrete-decision continuous-actuation control: balance of an inverted pendulum and pumping a pendulum swing". *Journal of Dynamic Systems, Measurement, and Control*, **137**(5), p. 051012.
- [19] Bhounsule, P. A., Cortell, J., Grewal, A., Hendriksen, B., Karssen, J. D., Paul, C., and Ruina, A., 2014. "Low-bandwidth reflex-based control for lower power walking: 65 km on a single battery charge". *International Journal of Robotics Research*, **33**, 10, pp. 1305 – 1321.
- [20] Hobbelen, D., and Wisse, M., 2007. "A disturbance rejection measure for limit cycle walkers: The gait sensitivity norm". *IEEE Transactions on robotics*, **23**(6), pp. 1213–1224.
- [21] Ameperosa, E., Miller, S., Seay, K., Ulep, R., and Bhounsule, P. A., 2015. Roadrunner: A rimless-wheel based legged robot. Youtube link <https://youtu.be/>

xTSG48KMMR4, July.

- [22] Cotton, S., Godowski, J. C., Payton, N. R., Vignati, M., Schmidt-Wetekam, C., Black, C., et al., 2015. Multi-legged running robot, Jan. 14. US Patent App. 14/596,514.
- [23] Smith, A. C., and Berkemeier, M. D., 1998. “The motion of a finite-width rimless wheel in 3d”. In *Robotics and Automation*, 1998. Proceedings. 1998 IEEE International Conference on, Vol. 3, IEEE, pp. 2345–2350.
- [24] Sabaapour, M., Hairi Yazdi, M., and Beigzadeh, B., 2015. “Passive turning motion of 3d rimless wheel: novel periodic gaits for bipedal curved walking”. *Advanced Robotics*, **29**(5), pp. 375–384.

Appendix A: Multimedia Extension

A video of the hardware prototype available on this YouTube link: <https://youtu.be/xTSG48KMMR4>.