

Design and control of a micro ball-balancing robot (MBBR) with orthogonal midlatitude omniwheel placement

Daniel Yang, Eric Sihite, Jeffrey M. Friesen, Thomas Bewley, *UCSD Coordinated Robotics Lab*¹

Abstract—Ball-balancing robots (BBRs) are endowed with rich dynamics. When properly designed and stabilized via feedback to eliminate jitter, and intuitively coordinated with a well-designed smartphone interface, BBRs exhibit a uniquely fluid and organic motion. Unlike mobile inverted pendulums (MIPs, akin to unmanned Segways), BBRs stabilize both fore/aft and left/right motions with feedback, and bank when turning. Previous research on BBRs focused on vehicles from 50cm to 2m in height; the present work is the first to build significantly smaller BBRs, with heights under 25cm. We consider the unique issues arising when miniaturizing a BBR to such a scale, which are characterized by faster time scales and reduced weight (and, thus, reduced normal force and stiction between the omniwheels and the ball). Two key patent-pending aspects of our design are (a) moving the omniwheels to contact the ball down to around 20 to 30 deg N latitude, which increases the normal force between the omniwheels and the ball, and (b) orienting the omniwheels into mutually-orthogonal planes, which improves efficiency. Design iterations were facilitated by rapid prototyping and leveraged low-cost manufacturing principles and inexpensive components. Classical successive loop closure control strategies are implemented, which prove to be remarkably effective when the BBR isn't spinning quickly, and thus the left/right and fore/aft stabilization problems decompose into two decoupled MIP problems.

I. INTRODUCTION

Ball-balancing robots (BBRs) exhibit rich 3D dynamics and are capable of fluid and graceful motion. Early research on BBRs appeared around 2006 (see [1]). To date, most BBR research has focused on human-scale designs ([1], [2], [3]), ball-balancing transportation vehicles ([4], [5], [6]), and knee-high to waist-high designs ([8]). These larger vehicles weigh between 8.7kg and 45kg, and are characterized by significantly higher build cost and slower dynamics than the designs considered here, which weigh 580g, are 22cm tall, and cost less than \$200 to build, with significant cost savings possible when mass produced. Since a micro ball-balancing robot (MBBR, pronounced “Ember”) is much cheaper to manufacture, it is viable for commercial applications in entertainment, service, education, and research.

There are significant challenges and limitations to contend with when attempting to miniaturize a BBR. Tolerances become more stringent and cross-sectional areas decrease, lowering the yield strength of mechanical components. Scaling down the characteristic length scale, l , of a given design generally reduces the volume and mass of the design by l^3 ; on a BBR, this significantly reduces the normal force between the omniwheels and the ball, creating problems with

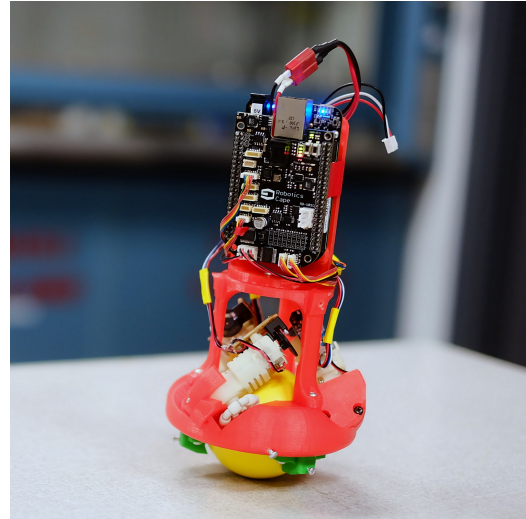


Fig. 1: Prototype micro ball-balancing robot (MBBR).

slip. Simultaneously, the time scale of the nonminimum-phase inverted-pendulum dynamics of a BBR is $\sqrt{l/g}$, so as l is reduced, the time scale decreases, and the actuators must respond faster, further exacerbating the slip problem. In extreme cases, the drive wheels may even lose contact with the ball completely for short periods of time.

In this paper, we address the design and control of an MBBR, addressing issues such as manufacturing tolerances, drivewheel slippage, and fast dynamics. We present several design iterations, including our patent-pending ideas of (a) “midlatitude” (close to the equator of the ball when the south pole is down) placement of omniwheels on the ball and (b) mutually-orthogonal orientation of the omniwheels. We begin with an outline of the design procedure and objectives. Next, we discuss the omniwheel placement and orientation. We then derive the decoupled equations of motion and describe the controller used to stabilize the MBBR under nonspinning conditions (a follow-up paper will focus specifically on the problem of control of a rapidly spinning MBBR). Finally, we present the experimental results of several MBBR prototypes, including our final MBBR designs.

II. PRELIMINARY DESIGNS

To begin the design process, we compiled an ordered list of design requirements: size, robustness, cost, and energy efficiency. With such an open design space, we began by examining existing large-scale designs which seemed fit for miniaturization. Previous work on BBRs fall into two main categories: those driven by an inverse mouseball mechanism, and those driven by three omniwheels. At first, neither

*This work was supported by WowWee Robotics.

¹emails: {dgyang, esihite, jfriesen, bewley}@ucsd.edu

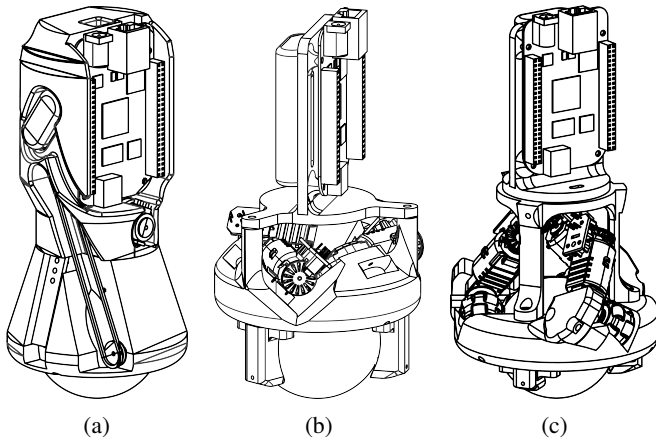


Fig. 2: Three MBBR mechanisms considered in this paper: a) belt-driven inverse mouseball mechanism, b) single-row omniwheel-driven mechanism with conventional omniwheel placement, and c) double-row omniwheel-driven mechanism with mutually-orthogonal omniwheel placement.

category presented obvious barriers to miniaturization. The many design choices available, the inaccuracy of analytical friction models, and various unresolved questions related to manufacturing led to the conclusion that rapid test and iteration were essential to converge quickly to the best practical solution. We thus decided to pursue both the inverse mouseball mechanism and the omniwheel-driven mechanism in parallel. Using low-cost additive manufacturing techniques, we iterated quickly on various design changes, and built more than 10 different working prototypes, three of which are shown in Fig. 2, in a span of three months. We successfully balanced both archetypes of the MBBR, eventually downselecting to the omniwheel-driven design.

The inverse mouseball mechanism (Fig. 2a) relies on two perpendicular rollers along the equator of the ball. A small, low-friction bearing is used at the top of the ball to support the weight of the upper body. Spring-loaded idler wheels at opposite points along the equator press the rollers against the ball to create enough friction to eliminate slip between the rollers and the ball. This allows the rollers to actuate the ball in the two horizontal directions independently.

There were two key problems with MBBRs driven by this mechanism. The first was contamination. During testing, dirt and rubber particles would build up on the support bearing, idler wheels, and drive rollers. This would increase friction and degrade performance to the point of failure. For large BBRs, the torques and forces involved are greater in magnitude, and tolerances are relaxed, so a thin layer of dirt has little effect on performance. However, for an MBBR, dirt buildup on the rotating components proved to be a critical point of failure.

The second problem encountered was the ball being pushed out of the socket during maneuvers. The rollers transmit torque to the ball by applying a friction force along the ball's equator. Depending on the direction of rotation, this force tends either to push the ball farther into the socket, or to pull the ball away from the socket. The only force

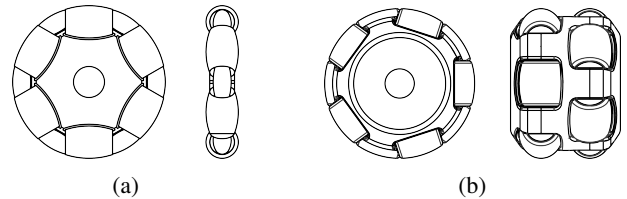


Fig. 3: Common omniwheels: a) single row, b) double row.

preventing the ball from leaving the socket is the weight of the robot itself. Since MBBRs have reduced mass, their weight is generally insufficient to keep the ball in the socket. Even if the ball does not leave the socket, the design suffers from asymmetric friction. As the roller actuates the ball in one direction, the ball is forced into the socket, increasing both the normal force and the friction at the top support bearing; when actuating the ball in the opposite direction, the friction at the top support bearing is reduced.

Additionally, the inverse mouseball mechanism does not control the yaw of the robot about its vertical axis, so additional actuators would be needed to make the robot face in a desired direction. Weighing all of these considerations, we ultimately concluded that the inverse mouseball mechanism, though feasible, was the lesser of the two available choices.

III. OMNIWHEEL-DRIVEN MBBR DESIGN

We now discuss the design challenges of building an omniwheel-driven MBBR. We begin by discussing the major limiting factor of such a design: the omniwheels themselves.

A. Omniwheel Design

An ideal omniwheel (see Fig. 3) has

- zero friction in the direction of its axis, which is achieved using rollers around its circumference,
- zero friction resisting yaw about the contact patch, and
- zero slip (infinite stiction) in its direction of rotation.

The smallest off-the-shelf omniwheels found were too large, at 4cm in diameter. We thus custom fabricated smaller omniwheels. Due to manufacturing tolerances and material strength limitations, the smallest omniwheels we could reliably manufacture were 2.5cm in diameter. Two omniwheel designs were selected for testing (see Fig. 3): the single-row omniwheel (SROW) design found in [9], [10], and the more conventional double-row omniwheel (DROW) design.

The SROW design considered (Fig. 3a) consists of 12 rollers lying in the same plane. The rollers alternate between large and small, allowing the smaller rollers to nest within the larger rollers to form an nearly circular profile. In practice, SROWs allowed for very smooth actuation of the ball, due to the absence of gaps between rollers and the single contact point on the ball. Several drawbacks were also encountered. First, the design complexity is high, with 37 parts per wheel, which increases the manufacturing cost significantly. Additionally, parts of the hub had to be made as thin as 1mm, which were prone to mechanical failures.

The DROW design considered (Fig. 3b) consists of 12 equally-sized rollers in two parallel planes, which greatly

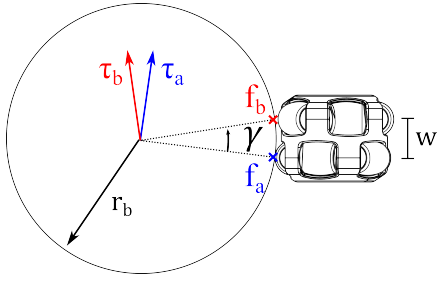


Fig. 4: Double row omniwheel and ball contact interaction.

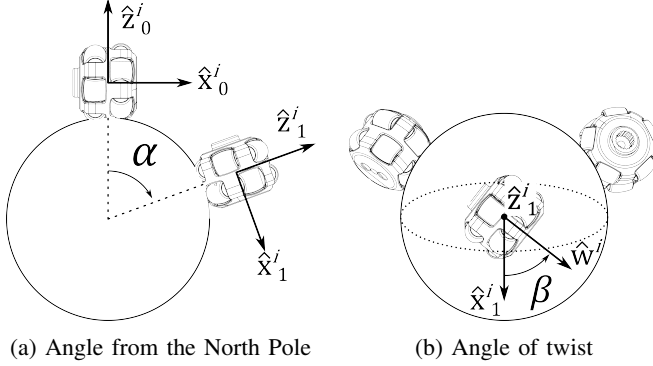


Fig. 5: Omniwheel placement and orientation

simplifies construction. Although the part count is still relatively high, the individual pieces are larger and more robust. The drawback of the DROW design lies in the differences in the engagement of the two different rows on the curved surface of the ball. On flat surfaces, a DROW behaves much like a SROW. However, in MBBRs, as the ratio between the ball diameter and the distance between the two rows of rollers becomes smaller, the omniwheels induce a characteristic “wobble” into the dynamics, as derived below, that is nearly impossible to eliminate.

As illustrated in Fig. 4, as the DROW rotates, the normal force f_a generated by the DROW on the ball creates (via stiction) a torque on the ball, τ_a , in a direction perpendicular to both f_a and the azimuthal direction of the omniwheel. As the omniwheel continues to rotate, the second roller contacts the ball, resulting in a normal force f_b and concomitant torque on the ball, τ_b . It is thus seen that the torque that the DROW generates on the ball switches back and forth between between τ_a and τ_b . This switching induces a periodic disturbance which leads to the undesirable “wobble”. The angle between these two axes of rotation is given by

$$\sin(\gamma) = w/(2r_b). \quad (1)$$

As the ratio between the radius of the ball and the distance between the two rows of rollers in the DROW increases, the wobble diminishes, as verified in §VI. In practice, we found both the SROW and DROW designs viable for miniaturization, though neither was perfect.

B. Midlatitude Omniwheel Placement

Following the work of others (see §I), our early omniwheel-driven MBBR prototypes, as seen in Fig. 2b,

placed the omniwheels between $\alpha = 30^\circ$ and $\alpha = 45^\circ$, and took $\beta = 0^\circ$ (see Fig. 5). With the omniwheels in such a position, we found that slippage between DROW and the ball was significant. To mitigate this problem, the friction between the wheel and the ball must be increased. There are two ways to accomplish this: increase the coefficient of friction between the ball and rollers, or increase the normal force between the omniwheels and the ball. Since there are limited options in materials, and it is undesirable to increase normal force by adding mass to the main body, we sought other solutions to this problem. We found that placing omniwheels closer to the equator of the ball (that is, in the “midlatitudes”), could greatly increase the normal force between the ball and omniwheels. It is easily seen that the normal force is equal to:

$$N = mg/[3 \cos(\alpha)] \quad (2)$$

The normal force monotonically increases as α increases from near 0° to near 90° . Thus, by moving the omniwheels from $\alpha = 45^\circ$ to 70° , we can double the normal force between the omniwheels and the without adding any mass. This simple geometric change significantly mitigated the omniwheel slippage issues. Note that, if α becomes too large, the amplified normal forces increase friction in the drivetrain, and degrades performance. We built several prototypes, with $\alpha = \{45^\circ, 60^\circ, 70^\circ, 80^\circ\}$, and found that, for our design, $\alpha = 60^\circ$ to 70° represented a good compromise.

The lower placement of the omniwheels also effectively solved the problem of the omniwheels losing contact with the ball during quick maneuvers. Lowering the omniwheels result in larger normal forces, which allows the omniwheels to grip the ball better, and the contact points upon which the upper body rests are spaced farther apart, requiring a larger moment to tip the upper body off the ball.

There is a significant drawback to lowering the omniwheel placement if the orientation of the omniwheels is left in the conventional orientation with $\beta = 0$. The magnitude of the component of the torque which contributes to the yaw of the ball about the vertical axis scales with $\sin(\alpha)$. Conversely, the magnitude of the components of torque which contribute to translational movement of the ball scales with $\cos(\alpha)$. Thus, as α approaches 90° , the actuator input corresponding to balancing the vehicle approaches 0, and balancing becomes impossible. A natural solution to this problem is achieved by varying β , as discussed below.

C. Orthogonal Omniwheel Orientation

We now consider the issue of the omniwheel orientation β . Our solution of mounting the omniwheels in mutually-orthogonal planes increases the overall efficiency of the power transfer and, as discussed above, is desirable when used in conjunction with midlatitude omniwheel placement. Traditionally (see, e.g., [7]), omniwheels are spaced evenly apart, with 120° separation, and oriented in a radially-symmetric fashion about the z-axis, with $\beta = 0^\circ$. By aligning the three omniwheels in mutually-orthogonal planes, we effectively decouple their effects.

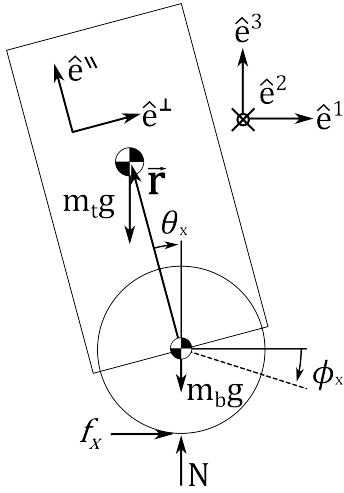


Fig. 6: Generalized coordinates of MBBR

If the omniwheels are *not* in mutually orthogonal planes, actuating one while holding the other two fixed results in both rolling of the ball about an axis τ parallel to that of the actuated omniwheel, and rolling about the other two axes, which gives no slip in the direction of rotation of the other two omniwheels. This is achieved by spinning the rollers of all three omniwheels, including the actuated omniwheel.

If, on the other hand, the omniwheels *are* in mutually orthogonal planes, actuating one while holding the other two fixed results in pure rolling of the ball about an axis τ parallel to that of the actuated omniwheel, and zero rolling about the other two axes. This is more direct and efficient, as it doesn't result in the spinning of the roller of the actuated omniwheel.

Our method to enforce orthogonal omniwheel orientation follows: let the torque produced by each motor on the omniwheel $i \in \{1, 2, 3\}$ be represented by $\tau^i = \tau_i \hat{\mathbf{w}}^i$, where the scalar τ_i is the magnitude of the torque, and $\hat{\mathbf{w}}^i = (\hat{w}_x^i, \hat{w}_y^i, \hat{w}_z^i)^T$ is the unit vector in the direction of the vector τ^i . The actuators τ_i contribute to the three components of torque in the upper body coordinates as follows

$$\begin{pmatrix} \tau_x \\ \tau_y \\ \tau_z \end{pmatrix} = \frac{r_b}{r_w} \sum_{i=1}^3 \tau^i = \frac{r_b}{r_w} \begin{pmatrix} \hat{\mathbf{w}}^1 & \hat{\mathbf{w}}^2 & \hat{\mathbf{w}}^3 \end{pmatrix} \begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix}, \quad (3)$$

$$T_w = (\hat{\mathbf{w}}^1 | \hat{\mathbf{w}}^2 | \hat{\mathbf{w}}^3), \quad (4)$$

where the r_b and r_w are the radius of the ball and omniwheel, respectively, the matrix T_w is invertible, and the torques τ_x , τ_y and τ_z represent the torques in the upper body coordinates. The ball rotation can also be computed using this transformation from the omniwheel rotation values measured by the encoders. Defining $\boldsymbol{\tau} = (\tau_x, \tau_y, \tau_z)^T$, $\boldsymbol{\tau}_w = (\tau_1, \tau_2, \tau_3)^T$, $\boldsymbol{\phi} = (\phi_x, \phi_y, \phi_z)^T$, and $\boldsymbol{\phi}_w = (\phi_1, \phi_2, \phi_3)^T$, we have

$$\boldsymbol{\tau} = (r_b/r_w) T_w \boldsymbol{\tau}_w, \quad (5)$$

$$\boldsymbol{\phi} = (r_w/r_b) T_w \boldsymbol{\phi}_w, \quad (6)$$

where ϕ_x , ϕ_y and ϕ_z are the ball rotation in the upper body coordinates, and ϕ_1 , ϕ_2 , ϕ_3 are the omniwheels rotational values as measured by the encoders.

TABLE I: Some orthogonal omniwheel orientations.

α	45°	60°	70°	80°
β	35.237°	48.186°	52.082°	54.145°

The omniwheels are orthogonal to each other if the vectors $\hat{\mathbf{w}}^1$, $\hat{\mathbf{w}}^2$, $\hat{\mathbf{w}}^3$ are mutually orthogonal. This may be achieved by calculating the $\hat{\mathbf{w}}^i$ given the α and β , then checking if $\hat{\mathbf{w}}^i \cdot \hat{\mathbf{w}}^j = 0$ for $i \neq j$ and $i, j \in \{1, 2, 3\}$. The $\hat{\mathbf{w}}^i$ can be determined using the following equations:

$$\hat{\mathbf{w}}^i = \text{Rot}(\hat{\mathbf{z}}_1^i, \beta) \text{Rot}(\hat{\mathbf{y}}_0^i, \alpha) \hat{\mathbf{x}}_0^i, \quad (7)$$

$$\hat{\mathbf{z}}_1^i = \text{Rot}(\hat{\mathbf{y}}_0^i, \alpha) \hat{\mathbf{z}}_0^i, \quad (8)$$

$$\hat{\mathbf{w}}^i = \text{Rot}(\hat{\mathbf{e}}^3, 2\pi/3)^{i-1} \hat{\mathbf{w}}^1, \quad (9)$$

where $\text{Rot}(\hat{\mathbf{x}}, \alpha)$ is the Euler rotation matrix about $\hat{\mathbf{x}}$, and $\{\hat{\mathbf{x}}_0^i, \hat{\mathbf{y}}_0^i, \hat{\mathbf{z}}_0^i\}$ are the initial wheel axis coordinates, which are set up as shown in Fig. 5. In particular, our prototype is set up such that the initial coordinates for the first wheel are the same as lab coordinates $\{\hat{\mathbf{e}}^1, \hat{\mathbf{e}}^2, \hat{\mathbf{e}}^3\}$. Given a particular value for α , we can determine the β which gives us $\hat{\mathbf{w}}^i \cdot \hat{\mathbf{w}}^j = 0$ for $i \neq j$ using an iterative method. Some of the $\{\alpha, \beta\}$ pairs resulting in orthogonal omniwheel orientation are given in Table I; the $\alpha = 70^\circ$, $\beta = 52.082^\circ$ case gives a transformation matrix T_w of:

$$T_w = \begin{pmatrix} 0.2102 & -0.7883 & 0.5781 \\ 0.7889 & -0.2124 & -0.5765 \\ -0.5775 & -0.5775 & -0.5775 \end{pmatrix} \quad (10)$$

As discussed previously, there are multiple benefits of such an orthogonal omniwheel orientation. First, the conventional omniwheel orientation at large α angles leads to diminished torque available to translate the ball. Further, the orthogonal omniwheel orientation leads directly to rotation of the ball in the actuated direction; actuation of a single omniwheel doesn't result in a force applied against the other two omniwheels in a manner that results in the spinning of the rollers of the actuated omniwheel, thereby providing a smoother application of torque.

IV. DYNAMICS

The MBBR dynamics are inherently 3D, complex, and nonlinear. They can be derived with Lagrangian dynamics using no slip constraints between the ball and the floor ($x = r_b \phi_y$, $y = -r_b \phi_x$) and ideal omniwheel assumptions [5]. In order to apply linear control, we simplify the problem significantly by assuming that the yaw rate (spin) of the MBBR (i.e., around the z axis) is negligible (i.e., $\dot{\phi} \approx 0$). Next, we linearized the θ_x and θ_y about small angles. This simplifies the equations of motion into two linear, decoupled, nonminimum phase Mobile Inverted Pendulum (MIP) problems in the x and y directions. Additional simplifying assumptions include: no slip between the ball and the floor ($x = r_b \phi_y$, $y = -r_b \phi_x$), and ideal omniwheels. We define θ as the upper-body tilt angle, and ϕ as the ball rotation angle in upper-body coordinates. Note that the angles in the y -direction affect the linear displacement in the x -direction, and vice-versa.

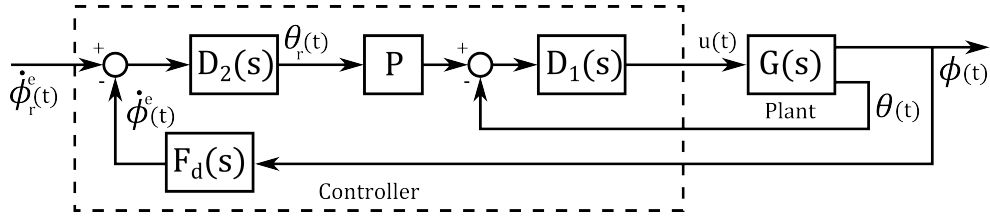


Fig. 7: Block diagram of SLC controller design.

Using the free body diagram in Fig. 6, we can write the planar equation of motion in the x-direction as

$$\begin{aligned} I_{b_y} \ddot{\phi}_y &= \tau_y - r_b f_x, \\ I_{t_y} \ddot{\theta}_y &= -\tau_y - P_z l \sin \theta_y - P_x l \cos \theta_y, \\ m_b \ddot{x} &= P_x - f_x, \\ m_t [\ddot{\mathbf{r}} \cdot \hat{\mathbf{e}}^1] &= m_t [\ddot{x} - l \cos \theta_y \ddot{\theta}_y + l \sin \theta_y \dot{\theta}_y^2] = -P_x, \\ m_t [\ddot{\mathbf{r}} \cdot \hat{\mathbf{e}}^\perp] &= m_t [\cos \theta_y \ddot{x} - l \ddot{\theta}_y], \\ &= -m_t g \sin \theta_y - P_z \sin \theta_y + P_x \cos \theta_y, \end{aligned}$$

where m is the mass, I is the moment of inertia about the center of mass, the subscripts b and t represents the 'ball' and 'top body' respectively, r_b is the ball radius, l is the distance from the top body center of mass to the ball center, g is the acceleration of gravity, x is the linear displacement of the ball, P_x and P_z are the normal forces, f_x is the friction between the ball and the floor, and τ_y is the net effective motor torque. Combining, applying the no slip condition, and linearizing using the small angle approximation on θ_y leads to the following linear system:

$$[I_{b_y} + (m_t + m_b)r_b^2] \ddot{\phi}_y + m_t r_b l \ddot{\theta}_y = \tau_y, \quad (11)$$

$$m_t r_b l \ddot{\phi}_y + I_{t_y}' \ddot{\theta}_y - m_t g l \theta_y = -\tau_y, \quad (12)$$

where I_{t_y}' is the inertia of the top body the ball's center of mass. The motor dynamics are augmented to the equations of motion above, resulting in a model that may be used to design the controller. The torque produced by the motors in this direction can be expressed using the linear motor model

$$\tau_y = (kV_{max}/R_m)u_y - (b + k^2/R_m)(\dot{\phi}_y - \dot{\theta}_y), \quad (13)$$

where k is the motor constant, V_{max} is the maximum battery voltage, R_m is the motor internal resistance, b is the motor Coulomb friction, and u_y is the effective motor duty cycle. The x-direction has similar dynamics, with a different inertia, ball rotation speed, and input directions.

V. CONTROLS

We use Successive Loop Closure (SLC) to separate the SIMO (single-input, multiple output) dynamics of each MIP problem into two SISO (single-input, single-output) parts. The (fast) inner loop stabilizes the upper body angle θ to some reference θ_r . The (slow) outer loop adjusts θ_r to stabilize the ball rotation angle ϕ about some reference value, which can be changed via remote control. This strategy works due to frequency separation (of at least a factor of 3) between the crossover frequencies of the inner and

outer loops. A block diagram representation of the controller formulation is given in Fig. 7. As discussed in, e.g., [11], the SIMO plant $G(s)$ can be idealized as a cascade of two SISO transfer functions, $G_1(s)$ and $G_2(s)$, as follows:

$$G_1(s) = \frac{\Theta_y(s)}{U_y(s)} = \frac{c_1 s}{a_3 s^3 + a_2 s^2 + a_1 s + a_0}, \quad (14)$$

$$G_2(s) = \frac{\Phi_y(s)}{\Theta_y(s)} = \frac{(b_2 s^2 + b_0)}{s^2}, \quad (15)$$

$$\begin{aligned} c_1 &= -(kV_{max}/R_m)(I_{b_y} + (m_t + m_b)r_b^2 + m_t r_b l), \\ a_3 &= I_{t_y}' [I_{b_y} + (m_t + m_b)r_b^2] - (m_t r_b l)^2, \\ a_2 &= (b + k^2/R_m)(I_{b_y} + (m_t + m_b)r_b^2 + 2m_t r_b l + I_{t_y}'), \\ a_1 &= -(I_{b_y} + (m_t + m_b)r_b^2)(m_t g l), \\ a_0 &= -(b + k^2/R_m)m_t g l, \\ b_2 &= -(m_t r_b l + I_{t_y}')/(I_{b_y} + (m_t + m_b)r_b^2 + m_t r_b l), \\ b_0 &= m_t g l/(I_{b_y} + (m_t + m_b)r_b^2 + m_t r_b l). \end{aligned}$$

We use a "dirty differentiator" filter $F_d(s)$ to estimate the value of $\dot{\phi}$ to be used in the velocity controller as follows:

$$\dot{\phi}_y^e(s) = \frac{s}{s + \omega_c} \Phi_y(s) = F_d(s) \Phi_y(s), \quad (16)$$

where ω_c is the cutoff frequency of the filter. The inner loop controller $D_1(s)$ takes the difference between the reference body angle θ_r and the current angle θ as an input and outputs the motor command value u . The outer loop controller $D_2(s)$ takes the difference between the reference ball angular velocity $\dot{\phi}_r^e$ and the current speed $\dot{\phi}$ and outputs the reference body angle θ_r . Each controller $D_1(s)$ and $D_2(s)$ are designed independently, where the prescaler P value is set to counteract the steady state error of the inner loop.

Using the parameters we measured and $\omega_c = 40$ rad/s, the system transfer functions are:

$$G_1(s) = \frac{17.9s}{(s - 13.33)(s + 13.45)(s + 0.06)}, \quad (17)$$

$$G_2(s) = \frac{-3.48(s + 8.63)(s - 8.63)}{s^2}, \quad (18)$$

$$F_d(s) = \frac{s}{s + 40}. \quad (19)$$

The transfer functions are the same for both x and y-directions as long as their respective inertia are the same.

We then use the following controller for both directions:

$$D_1(s) = 50 \frac{(s + 13.45)(s + 0.06)}{s(s + 40.34)} \quad (20)$$

$$D_2(s) = 4P \frac{1}{(s + 8.63)} \quad (21)$$

$$P = 1/2.5; \quad (22)$$

These controllers have a closed loop step response with inner and outer loop rise time t_r of 0.11 sec and 0.38 sec respectively. The inner loop has 5% overshoot while the outer loop has no overshoot.

We assumed that the heading ϕ_z is approximately constant, so we control the headings using the simple PD controller

$$u_z = K_p(\phi_{zref} - \phi_z) + K_d(\dot{\phi}_{zref} - \dot{\phi}_z). \quad (23)$$

After determining u_x , u_y , and u_z , we convert these values back into each individual motor PWM commands u_i , $i = \{1, 2, 3\}$, using the same transformation for τ in (5) because of the linear relationship between τ and u . Then we calculate

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \frac{r_w}{r_b} T_w^{-1} \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}. \quad (24)$$

After the conversion, we also need to take into account the static friction on the omniwheels. Using midlatitude omniwheel placement introduces more normal force into the system, increasing friction. We used the nonlinear input value given by:

$$u'_i = c \operatorname{sign}(u_i) + (1 - c)u_i \quad (25)$$

where c is the static boost constant and $u'_i \in [-1, 1]$ is the final motor command value for the motor i .

A. Mechanical Builds

All of the prototypes consisted of similar mechanical architectures and hardware. For the sake of discussion we will look at our most robust configuration: the DROW driven MBBR with wheels placed at $\alpha = 70^\circ$.

One of the most difficult components to source was the main ball, identified as (4) in Fig. 8. It was found that the ball must be sufficiently round and have a high coefficient of friction. Most balls smaller 7cm in diameter had surface imperfections and lacked the roundness to work properly. Also, balls that had a high coefficient of friction tended to be too compliant, causing the ball to deform under large dynamic loads. We found that a high quality rubber juggling ball was the best compromise between hardness, roundness, and surface friction. The omniwheels were placed in contact with the ball in the orthogonal configuration with $\alpha = 70^\circ$ and $\beta = 52.082^\circ$. The motors (2) and omniwheels (1) were rigidly mounted to a 3D printed toroidal chassis (5). The torus was designed to remain rigid under the larger normal forces associated with large α angles. Low friction acetal support balls (6) were used to constrain the ball against the omniwheels under dynamic maneuvers. We designed the support ball housing to be adjustable, to allow varying of the preload of the support ball against the ball. Finally, 3D

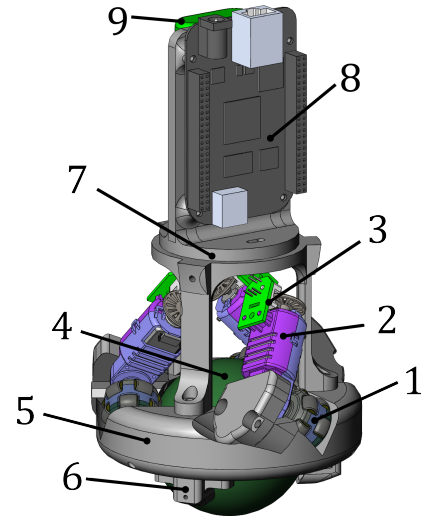


Fig. 8: MBBR prototype consisting of 1) 2.5cm DROW, 2) toy-grade plastic gearmotor, 3) optical encoder, 4) 6.4cm diameter rubber ball, 5) 3D printed chassis, 6) support ball bearing housing with acetal ball, 7) 3D printed electronics mount, 8) BeagleBone Black microprocessor, and 9) Panasonic 18650 lithium-ion battery pack.

printed structures (7) were used to mount the battery (9) and electronics (8).

VI. EXPERIMENTAL RESULTS

After many design iterations, we observed several trends that were consistent with our analysis. As expected, SROW resulted in lower friction and better overall performance due to their roundness and single contact plane. However, we encountered issues with SROW durability which we believe are intrinsic to the SROW design at small scales. Also, by increasing the α angle, we were able to stabilize designs that were not viable due to ball contact issues. For example, our DROW MBBR designs only worked if α was increased from 45° to 70° . However, increasing α to 80° significantly increased friction and could jam the ball into the socket due to the combination of widely spaced contact points and ball compliance. In addition, we discovered some designs were more reliant on the support balls to constrain the ball. For example, with the support balls removed, we found the SROW $\alpha = 70^\circ$ design was able to balance but the SROW $\alpha = 45^\circ$ could not. In practice, all designs benefited from supporting balls, however, our experiments demonstrated that larger α angles can constrain the ball better.

We now present data from three successful MBBR designs (see Fig. 2): conventional SROW $\alpha = 45^\circ$, orthogonal SROW $\alpha = 70^\circ$, and orthogonal DROW $\alpha = 70^\circ$. Data was logged from these three prototypes during a 3 second velocity input of 1.2 rad/s which commands the robot forewards and then to come to a full stop, shown in Fig. 9. We see that all three prototypes respond and track the velocity step well and that the actual angle θ closely matches the controller's inner loop reference θ_r . This allows the prototypes to be smoothly driven using remote control.

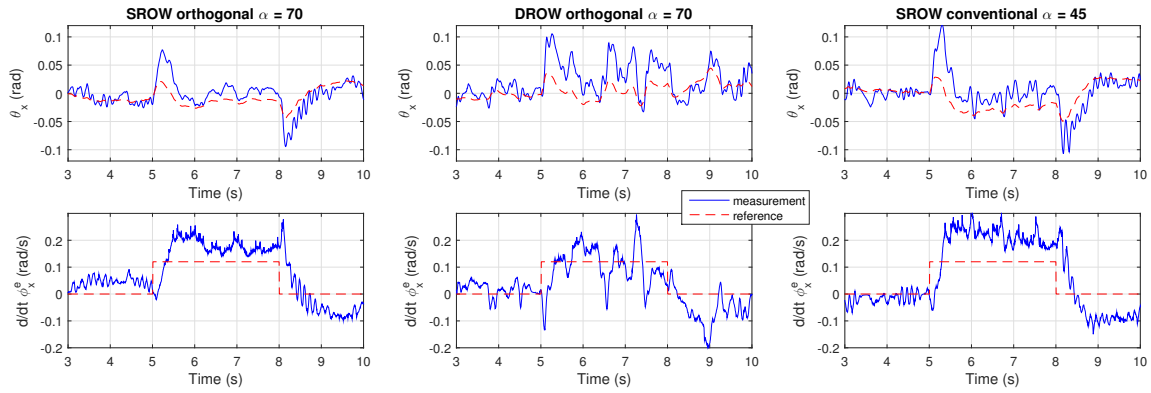


Fig. 9: Plot of the θ_x and $\dot{\phi}_x^e$ response of the prototypes.

Looking at the θ values of the DROW MBBR in Fig. 9, it is seen to suffer from a ~ 8 Hz wobble caused by the reasons outlined in § III. In addition, the $\dot{\phi}$ plot shows that the prototype struggles to follow the velocity step response. This is because the inner loop, which has priority over the outer loop, attempts to stabilize the wobble prior to following the velocity command. Also, the controller gains for the DROW MBBR needed to be increased relative to the SROW MBBR in order to achieve similar performance. We believe this is caused by the switching of actuation planes, which needs additional control to counteract the perturbations. The squared sum of the control values at time step k , $\mathbf{u}_k$ for $t = [3, 10]$ sec can be calculated to compare the control effort J_u of both designs. The J_u for SROW and DROW are 430 and 1050 respectively, which means that the SROW can be driven much more efficiently than DROW. Comparing the SROW build with $\alpha = 45^\circ$ and $\beta = 0^\circ$ to the SROW build with $\alpha = 70^\circ$ and orthogonal omniwheels, we saw a relatively small difference in performance.

VII. CONCLUSIONS

From our analysis and experimentation, we showed that the SROW exhibited less slip against the ball and had smoother actuation compared to the DROW. However SROWS were more fragile, and might be cost prohibitive in some applications. Also, increasing the α angle can counteract the low mass of MBBRs by increasing the normal force to eliminate omniwheel slip and reduce ball ejection. However, α should be increased carefully to avoid excessive friction and stress on the omniwheels. Ultimately, the increased efficiency of the orthogonal configuration turned out to be relatively minor in our initial testing. Note, however, that the orthogonal orientation is essential when larger α angles are employed in order to maintain sufficient torque to balance the vehicle. Note that the ideas discussed in this paper can also be applied to high-performance large-scale BBRs.

All of the MBBR prototypes presented here exhibited instabilities when trying to translate and rotate simultaneously. The use of planar dynamic models (one for fore/aft, one for left/right) is only justified when the body is essentially not rotating ($\dot{\phi}_z \approx 0$). When the body is rotating at a significant rate, the fore/aft and left/right dynamics are coupled, and the

controllers presented in this paper will fail to stabilize the system. A future paper will discuss our ongoing investigation in the development of a new controller, based on a fully 3D dynamic model, which is effective even when the MBBR is rotating quickly.

ACKNOWLEDGMENT

The authors thank Adam Fairless, Davin Sufer, Peter Yanofsky, and Cheung Wai On at WowWee Robotics for their collaborative work with us on this project, and for teaming with us to develop compelling new robotic systems, and to deliver them market. The authors also thank David Gibbons at the UCSD Tech Transfer office for bringing the UCSD and WowWee teams together, and for pursuing vigorous patent protection on the unique designs developed.

REFERENCES

- [1] T.B. Lauwers, G.A. Kantor, R.L. Hollis, "A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, pp. 2884-2889, May 2006.
- [2] U. Nagarajan, G. Kantor, R. Hollis, "The ballbot: An omnidirectional balancing mobile robot," in *The Int'l Journal of Robotics Research*, 2013.
- [3] A. Lotfiani, M. Keshmiri, M. Danesh, "Dynamic analysis and control synthesis of a spherical wheeled robot (Ballbot)," in *RSI/ISM Int'l Conf. on Robotics and Mechatronics*, 2013.
- [4] C. Tsai, C. Chan, L. Kuo, "LQR motion control of a ball-riding robot," in *IEEE/ASME Int'l Conf. on Advanced Intelligent Mechatronics*, pp. 861-866, July 2012.
- [5] T. Hoshino, S. Yokota, T. Chino, "OmniRide: A personal vehicle with 3 DOF mobility," in *Int'l Conf. on Control, Automation, Robotics and Embedded Systems*, Dec 2013.
- [6] C. Souleymane, C. Tsai, Y. Chiu, "Self-balancing control using Wavelet Fuzzy CMAC for uncertain omnidirectional ball-driven vehicles," in *IEEE Int'l Conf. on Fuzzy Theory and Its Applications*, 2013.
- [7] M.J. Bjrenstam, M. Lennartsson, "Development of a ball balancing robot with omni wheels," M.S. thesis, Dept. of Automatic Control, Lund University, Lund, Sweden, 2012.
- [8] M. Akbari, H.Z. Kheibari, A.S.M. Nejad, "Timing belt gearbox in Ballbot robot," in *RSI/ISM Int'l Conf. on Robotics and Mechatronics*, 2013.
- [9] M. Kumagai, T. Ochiai, "Development of a Robot Balanced on a Ball - First Report, Implementation of the Robot and Basic Control," in *Journal of Robotics and Mechatronics*, vol.22 no.3, 2010, pp 348-355.
- [10] L. Hertig, D. Schindler, M. Bloesch, C.D. Remy, R. Siegwart, "Unified State Estimation for a Ballbot," in *IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pp. 2471,2476, May 2013.
- [11] T. Bewley, "Numerical Renaissance: simulation, optimization, and control," in press, 2015.