One is Enough!

Tom Lauwers, George Kantor, and Ralph Hollis

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
{*tlauwers,kantor,rhollis*}*@ri.cmu.edu*

Summary. We postulate that multi-wheel statically-stable mobile robots for operation in human environments are an evolutionary dead end. Robots of this class tall enough to interact meaningfully with people must have low centers of gravity, overly wide bases of support, and very low accelerations to avoid tipping over. Accordingly, we are developing an inverse of this type of mobile robot that is the height, width, and weight of a person, having a high center of gravity, that balances dynamically on a *single spherical wheel*. Unlike balancing 2-wheel platforms which must turn before driving in some direction, the single-wheel robot can move directly in any direction. We present the overall design, actuator mechanism based on an inverse mouse-ball drive, control system, and initial results including dynamic balancing, station keeping, and point-to-point motion.

1 Motivation

A significant, but frequently overlooked problem is that statically-stable wheeled mobile robots can easily become dynamically *unstable*. If the center of gravity is too high, or the robot accelerates/decelerates too rapidly, or is on a sloping surface, the machine can tip over. A robot must be tall enough to be able to interact with people and the human environment at a reasonable height. On the other hand, it must be skinny enough to easily make its way around without bumping into things or getting into peoples' way.

What is needed are robots that are safe; agile and capable of graceful motion; slender enough to easily maneuver in cluttered, peopled environments; and which readily yield when pushed around. It is surmised that intelligent machines of this sort can only be achieved with *dynamic stability*. This idea follows the model of humans and other animals which are intrinsically dynamically stable.

2 Background

A two-wheeled robot with inverse pendulum control developed in Japan was demonstrated in 1994 [2]. The two-wheeled design eliminated the need for a third castor-

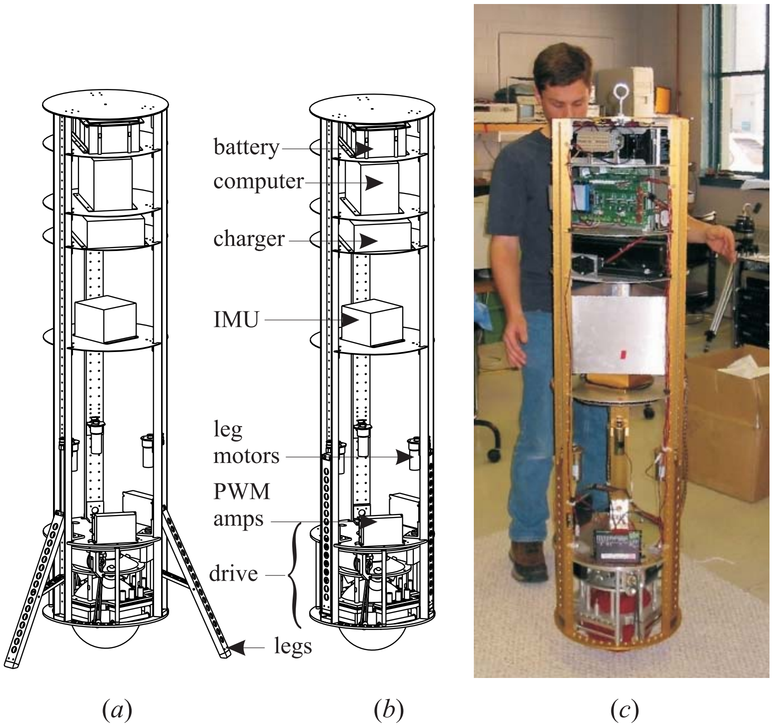


Fig. 1. Ballbot design and realization: (*a*) with three legs deployed, (*b*) with legs retracted into body, (*c*) balancing and station keeping.

ing wheel. The same group introduced a one-wheel balancing robot [3]. The wheel is a prolate ellipsoid like a rugby ball and is driven with an axle along the major axis. The body of the robot has a hinge above and perpendicular to this axis. The robot balances in the forward/backward directions by application of wheel torque in the manner of the two-wheeled design, and balances from side to side by leaning the body left or right at the actuated hinge. Recently, balancing wheel chairs[1] and balancing 2-wheel "Segway personal mobility devices"[2] have been introduced. The 2-wheel RMP robotic platforms [4] based on the Segway are the subject of much recent development in robotic locomotion.

The previous work on dynamically-stable rolling machines provides inspiration for our current research, yet is distinctly different. For example, there is no previous work proposing a balancing rolling machine whose body is supported by a single omni-directional spherical wheel. The previous rolling/balancing machines cannot immediately drive in a given direction without first re-orienting the drive mechanism. For example, a two-wheel balancing machine such as the Segway RMP cannot maneuver in tight spaces by moving sideways; a robot based on such a machine could

[1] Independence Technology, `http://www.indetech.com`.

[2] Segway human transporter, `http://www.segway.com.`

not open and close a door without knowing the precise location of the hinges in order to establish the correct turning radius. The rugby-ball robot cannot turn in place, but can only turn in a wide arc.

3 System Description

Ballbot, shown in Fig. 1, is a reconfigurable research platform developed and constructed to validate the notion of a dynamically stable robot resting atop a single, spherical drive wheel. It was designed to meet two goals: approximate the dimensions of a human being, and create a platform that is easily reconfigured for various present and future research efforts. The body is a cylinder 1.5 m tall, with a diameter of 400 mm and a weight of 45 kg. Three aluminum channels, held together by circular decks, define the structure of Ballbot's body. Three retractable landing legs are attached to the lower third of the channels, which when deployed allow Ballbot to remain standing after being powered down. Active components, such as computing, power, and sensing, are mounted on the decks, allowing these elements to be placed at varying positions along Ballbot's axis. Figures $1(a)$ and (b) show the design and Fig. $1(c)$ shows its present configuration successfully balancing and station keeping.

Ballbot is designed to be entirely self-contained; power is supplied by a 48V lead acid battery with operating time of several hours, and computing is performed onboard by a 200 MHz Pentium processor. Communication with Ballbot is through an 802.11b wireless link. A Crossbow Technology VG700CA-200 Inertial Measuring Unit (IMU) emulating a vertical gyro provides Kalman-filtered pitch and roll angles and rates with respect to gravity. The drive motors are connected to Copley Model 412 PWM amplifiers, with 1024 cpr encoders feeding motor shaft position back to the computer. Additionally, 1024 cpr encoders are placed on the passive rollers to measure ball rotation. The IMU and encoders provide all data required for full-state feedback control.

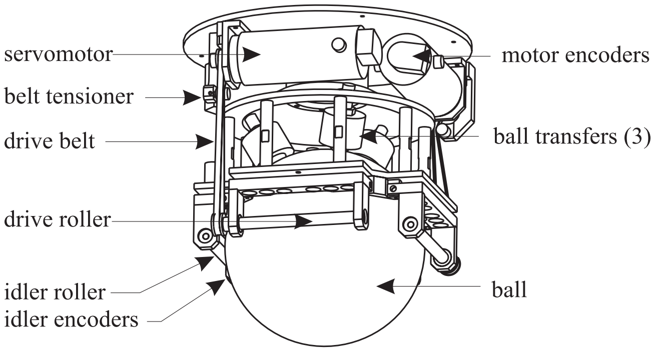


Fig. 2. Ballbot inverse mouseball drive mechanism

The drive mechanism, shown in Fig. 2, is essentially the inverse of a mouse ball drive: instead of the mouse ball driving the mouse rollers to provide computer input, rollers drive the ball to produce motion. The ball is a 200 mm diameter hydroformed steel shell covered with a 3.2 mm thick urethane outer layer. We have fabricated balls with urethane formulations of several different durometers. The ball is actuated by a pair of 12.7 mm diameter smooth stainless steel rollers placed orthogonally at the sphere's equator. These rollers are linked through timing belts to two high torque DC servomotors. Opposite the drive rollers are two spring-loaded passive idler rollers that apply force at the ball's equator to maintain contact between the drive rollers and the ball. This arrangement represents a compromise since some slippage is always present. For example, if one roller is being driven, the orthogonal roller must be slipping. This simultaneously demands both a high-friction and low-friction material for the ball. On the other hand, it is always desirable to have high friction between the ball and the floor. The drive works well but a fair amount of ball wear has been experienced, and we are at present still seeking a satisfactory compromise solution. The entire drive mechanism is attached to the body with a large diameter thrust bearing, allowing a third actuator (currently not installed) to re-orient the body in yaw. Finally, the entire Ballbot body rests on top of the ball on three commercial low friction, omni-directional ball transfer devices.

4 Simplified Ballbot Model

For the purposes of developing a stabilizing controller, we introduce and derive equations of motion for a simplified model of Ballbot. In this model, the Ballbot ball wheel is a rigid sphere, the body is rigid, and the control inputs are torques applied between the ball and the body. There is no slip between the wheel and the floor. Friction between the wheel and the floor and between the wheel and the body is modeled as viscous damping. Further, we assume that the motion in the median sagital plane and median coronal plane is decoupled and that the equations of motion in these two planes are identical. As a result, we can design a controller for the full 3D system by designing independent controllers for the two separate and identical planar systems.

Figure 3 is a diagram depicting the planar model. The Lagrangian formulation is used to derive the nonlinear equations of motion for the simplified model (see, *e.g.*, [1]). The first step is to compute the kinetic energy K_b of the ball:

$$K_b = \frac{I_b\dot{\theta}^2}{2} + \frac{m_b(r_b\dot{\theta})^2}{2},$$

where I_b, m_b, and r_b are, respectively, the moment of inertia, mass, and radius of the ball. The potential energy of the ball is $V_b = 0$. The kinetic energy K_B and potential energy V_B of the body are

$$K_B = \frac{m_B}{2}\left(r_b^2\dot{\theta}^2 + 2r_b\ell(\dot{\theta}^2 + \dot{\theta}\dot{\phi})\cos(\theta+\phi) + \ell^2(\dot{\theta}+\dot{\phi})^2\right) + \frac{I_B}{2}(\dot{\theta}+\dot{\phi})^2,$$

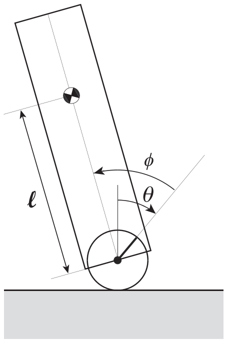


Fig. 3. Planar simplified Ballbot model used for controller design.

$$V_B = m_B g \ell \cos(\phi + \theta),$$

where I_B is the moment of inertia of the body about the center of the ball, ℓ is the distance between the center of the ball and the center of mass of the body, m_B is the mass of the body, and g is the acceleration due to gravity. The total kinetic energy is $K = K_b + K_B$ and the total potential energy is $V = V_b + V_B$.

Define the system configuration vector $q = [\theta \ \ \phi]^T$. The Lagrangian $\mathcal{L}$ is a function of q and $\dot{q}$ and is defined to be $\mathcal{L}(q, \dot{q}) = K - V$.

Let τ be the the component of the torque applied between the ball and the body in the direction normal to the plane. To model the viscous friction terms, define the vector

$$D(\dot{q}) = \begin{bmatrix} \mu_\theta \dot{\theta} \\ \mu_\phi \dot{\phi} \end{bmatrix},$$

where μ_θ and μ_ϕ are the viscous damping coefficients that model ball–ground and ball–body friction, respectively. Using this notation, the Euler-Lagrange equations of motion for the simplified Ballbot model are

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\partial \mathcal{L}}{\partial q} = \begin{bmatrix} 0 \\ \tau \end{bmatrix} - D(\dot{q}).$$

After computing the derivatives in the Euler-Lagrange equations and rearranging terms, the equations of motion can be expressed as

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) + D(\dot{q}) = \begin{bmatrix} 0 \\ \tau \end{bmatrix}. \tag{1}$$

The mass matrix $M(q)$ is

$$M(q) = \begin{bmatrix} \Gamma_1 + 2m_B r_b \ell \cos(\theta + \phi) & \Gamma_2 + m_B r_b \ell \cos(\theta + \phi) \\ \Gamma_2 + m_B r_b \ell \cos(\theta + \phi) & \Gamma_2 \end{bmatrix},$$

where

$$\begin{aligned}\Gamma_1 &= I_b + I_B + m_b r_b^2 + m_B r_b^2 + m_B \ell^2,\\ \Gamma_2 &= m_B \ell^2 + I_B.\end{aligned}$$

The vector of Coriolis and centrifugal forces is

$$C(q,\dot{q}) = \begin{bmatrix} -m_B r_b \ell \sin(\theta+\phi)(\dot{\theta}+\dot{\phi})^2 \\ 0 \end{bmatrix}$$

and the vector of gravitational forces is

$$G(q) = \begin{bmatrix} -m_B g \ell \sin(\theta+\phi) \\ -m_B g \ell \sin(\theta+\phi) \end{bmatrix}.$$

To put these equations into standard nonlinear state space form, define the state vector to be $x = [\, q^T \;\; \dot{q}^T \,]^T$ and define the input $u = \tau$. This together with Eq. 1 yields

$$\dot{x} = \begin{bmatrix} \dot{q} \\ M(q)^{-1} \left(\begin{bmatrix} 0 \\ u \end{bmatrix} - C(q,\dot{q}) - G(q) - D(\dot{q}) \right) \end{bmatrix} \triangleq f(x,u).$$

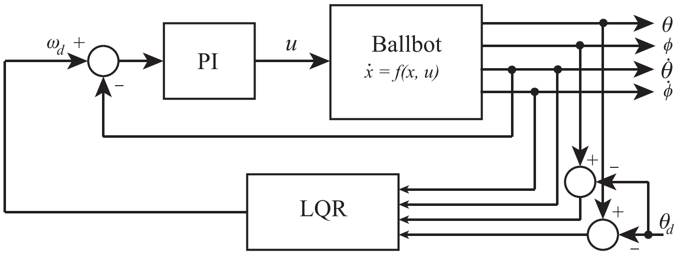


Fig. 4. Structure of stabilizing linear feedback controller.

5 Stabilizing Feedback Controller

The linear controller used to stabilize Ballbot has two loops: an inner loop that feeds ball velocity $\dot{\theta}$ back into a PI controller, and an outer loop linear quadratic regulator (LQR) that uses full state feedback. This architecture is shown in Fig. 4. The proportional gain k_p and integral gain k_i in the PI controller are chosen and experimentally tuned so that the actual ball velocity $\dot{\theta}$ tracks the desired ball velocity ω_d. The integral term adds an extra state to the system. Define the augmented state vector $x_a = [\, x^T \quad x_5 \,]^T$. The closed loop equations of motion of the inner loop can then be written as

$$\dot{x}_a = \begin{bmatrix} f\left(x, k_p(\omega_d - \dot{\theta}) + k_i(x_5 - \theta)\right) \\ \omega_d \end{bmatrix} \triangleq f_a(x_a, \omega_d).$$

The outer loop is designed by linearizing the inner loop equations of motion and applying LQR. Note that the simplified Ballbot system is at equilibrium whenever $\sin(\theta + \phi) = 0$ and $\dot{\phi} = \dot{\theta} = 0$. The objective is to design a controller that will balance Ballbot with the body straight up and hold it in a fixed position $\theta = 0$, which is equivalent to stabilizing the equilibrium point at $x_a = 0$. We begin by linearizing the equations of motion about this point:

$$\dot{x}_a = \underbrace{\left.\frac{\partial f_a}{\partial x_a}\right|_{x_a=0,\omega_d=0}}_{A} x_a + \underbrace{\left.\frac{\partial f_a}{\partial \omega_d}\right|_{x_a=0,\omega_d=0}}_{B} \omega_d.$$

Working out the partial derivatives yields

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \multicolumn{5}{c}{M_*^{-1} \begin{bmatrix} -m_B g\ell & -m_B g\ell & \mu_\theta & 0 & 0 \\ -m_B g\ell - k_i & -m_B g\ell & -k_p & \mu_\phi & k_i \end{bmatrix}} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ 0 \\ M_*^{-1} \begin{bmatrix} 0 \\ k_p \end{bmatrix} \\ 1 \end{bmatrix},$$

where M_* is simply the mass matrix $M(q)$ evaluated at $\theta = \phi = 0$.

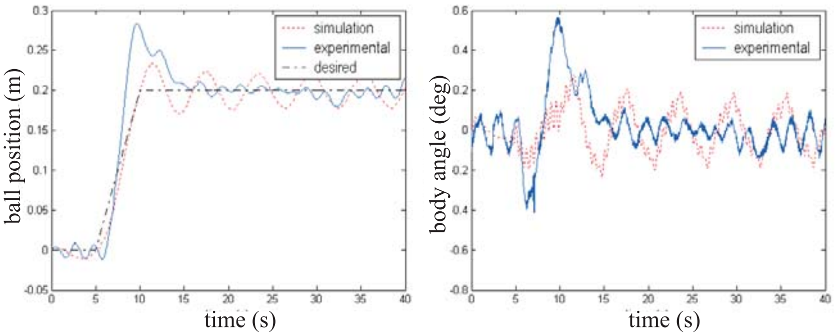
Now LQR can be used to find a linear state feedback controller that stabilizes the system about $x_a = 0$ and minimizes the cost function

$$J = \int (x_a(t)^T Q x_a(t) + R\omega_d(t)^2) dt.$$

We choose the structure of Q to be

$$Q = \begin{bmatrix} \gamma_b + \gamma_B & \gamma_B & 0 & 0 & 0 \\ \gamma_b & \gamma_B & 0 & 0 & 0 \\ 0 & 0 & \gamma_{\dot{b}} + \gamma_{\dot{B}} & \gamma_{\dot{B}} & 0 \\ 0 & 0 & \gamma_{\dot{B}} & \gamma_{\dot{B}} & 0 \\ 0 & 0 & 0 & 0 & \gamma_5 \end{bmatrix},$$

where γ_b, γ_B, $\gamma_{\dot{b}}$, $\gamma_{\dot{B}}$, and γ_5 can be loosely thought of as controlling the relative convergence rates of the ball angle, body angle, ball angular velocity, body angular velocity, and x_5, respectively. In practice, these parameters were hand tuned based on simulation results. For a given choice of Q and R, Matlab's LQR command can be



(a) (b)

Fig. 5. Moving between two locations in a straight line: (a) ball position in meters, (b) body angle in degrees.

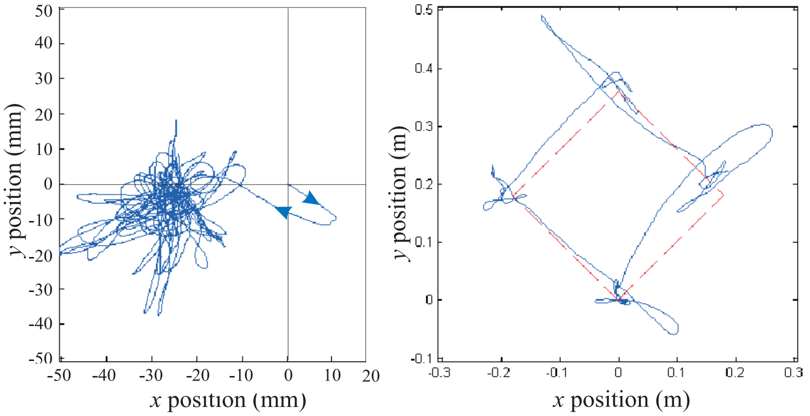
used to compute the associated gain matrix K, which defines the stabilizing feedback control law $\omega_d = -Kx_a$.

When implementing the controller on the actual robot, we were forced to deviate slightly from the controller presented above. We found that there is a practical limit on the magnitude of the gain k_4 that multiplies $\dot{\phi}$. Exceeding this limit induces an oscillation not present in the simplified Ballbot model. We hypothesize that this oscillation is due to flexibility in the body frame and the mechanics of the soft urethane layer that couples the drive roller to the ball. The K matrix generated by the LQR algorithm gives a k_4 that exceeds the practical limit, so we manually adjusted k_4 to an allowable level. Unfortunately, with this limit on k_4, it is not possible to directly stabilize Ballbot, which explains the need for the inner PI loop. Also, the gain k_5 turns out to be negligible, so it is set to zero in the experiments.

6 Initial Results

A number of tests were conducted to characterize physical system performance, and to make comparisons with simulation. During operation on a hard tiled floor, it was found that the machine was able to balance robustly, strongly resisting attempts to tip it over when a person applied torques to the body. However, it was not able to simultaneously balance and station keep. When operated on a carpeted surface, Ballbot was able to do both, presumably due to the extra damping afforded by the carpet material.

In the test run shown in Fig. 5, Ballbot was commanded to move from a starting position in a straight line to a goal position. There is an initial retrograde ball motion causing the body to lean toward the goal position, followed by a reverse motion to stop at the goal. As mentioned in the previous section, differences between simulation and experiment might derive from unknown frictional and spring forces. The divergence when station keeping is at most about 40 mm in position, and 0.5° in tilt.



(*a*) (*b*)

Fig. 6. Plots of the ball path during (*a*) balancing and station keeping, and (*b*) attempting to move in a square.

To see the typical motion jitter experienced during operation, one may plot the paths taken as the ball moves around on the carpeted floor. Figure 6(*a*) shows data taken from a 99 s run where Ballbot was released slightly out of balance, which was rapidly corrected by ball motion, followed by station keeping within a roughly circular region of about 40 mm diameter. Figure 6(*b*) shows Ballbot's attempt to track a square trajectory.

7 Discussion

Our results are preliminary and there is much that remains to refine Ballbot's model and control. Nevertheless, it would appear that Ballbot and its progeny might well represent the vanguard of a new type of wheeled mobile robot capable of agile, omnidirectional motion. Such robots, combined with the research community's ongoing work in perception, navigation, and cognition, could yield truly capable intelligent mobile robots for use in physical contact with people. If realizable and economically viable, they might well function as aids to elderly or disabled persons; provide guidance and assistance in public spaces; help with education and entertainment; perform domestic cleaning and housekeeping; or fetch and carry everyday objects. The more immediate goal of our research is simply to gain a deeper understanding of how such dynamic agility can be achieved in mobile machines interacting with people and operating in normal home and workplace environments.

Acknowledgment

This work was supported in part by NSF grant IIS-0308067.

References

1. A.M. Bloch. *Nonholonomic Mechanics and Control*. Springer, 2003.
2. Y.-S. Ha and S. Yuta. Trajectory tracking control for navigation of self-contained mobile inverse pendulum. In *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pages 1875–1882, 1994.
3. R. Nakajima, T. Tsubouchi, S. Yuta, and E. Koyanagi. A development of a new mechanism of an autonomous unicycle. In *Proc. IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, pages 906–12, Grenoble, France, September 7-11 1997.
4. H. G. Nguyen, J. Morrell, K. Mullens, A. Burmeister, S. Miles, N. Farrington, K. Thomas, and D. Gage. Segway robotic mobility platform. In *SPIE Proc. 5609: Mobile Robots XVII*, Philadelphia, PA, October 2004.