

A Dynamically Stable Single-Wheeled Mobile Robot with Inverse Mouse-Ball Drive

T. B. Lauwers, G. A. Kantor, and R. L. Hollis

The Robotics Institute

Carnegie Mellon University

Pittsburgh, Pennsylvania, USA

tlauwers@andrew.cmu.edu, kantor@ri.cmu.edu, rhollis@cs.cmu.edu

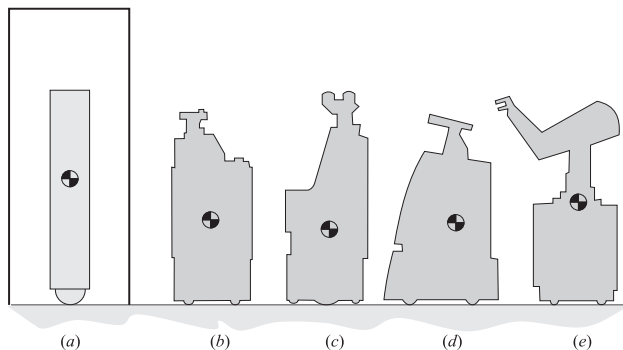


Fig. 1. Ballbot (this paper) and representative statically-stable mobile robots (silhouettes and centers of mass are approximate): (a) Ballbot [8], (b) Xavier [11], (c) Nursebot [2], (d) Minerva [12], (e) Romeo or Juliet [7]. The rectangle represents the approximate relative size of a house doorway.

Abstract—Multi-wheel statically-stable mobile robots tall enough to interact meaningfully with people must have low centers of gravity, wide bases of support, and low accelerations to avoid tipping over. These conditions present a number of performance limitations. 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.

I. 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.

Fig. 1 is a rough illustration of some venerable statically-stable robots currently in use in several research projects. The rapid development of each one of these example robots (and others which could be cited) was made possible by the recent introduction of more-or-less standardized commercial robot bases. In fact, this has been a tremendous boon to researchers developing mobile robots capable of meaningful

interactions with people—permitting the mobility issue to be safely “ignored,” while enabling workers to focus on their respective application domains. (In earlier days, robot builders by necessity were embroiled with the details of locomotion—often for years—before moving on to the research topic of interest.)

Fig. 2(a) depicts the case for a three-wheeled omnidirectional base capable of omni-directional motion. The robot radius is r_1 , the effective wheel base is defined by $r_2 < r_1$, and the (surprisingly small) tipping moment arm is shown as d . The four-wheel, generally nonholonomic base shown in Fig. 2(b) has a somewhat greater tipping moment arm, but suffers from the fact that some form of suspension must be incorporated to ensure that all four wheels remain in contact with the floor. In the side view of Fig. 2(c), the robot is moving to the left and decelerating with wheel braking forces F_{b1} and F_{b2} acting on the body. In this case, $F_1 + F_2 = mg$, $F_1 d = F_2 d + (F_{b1} + F_{b2})h$, and the acceleration is $a = (F_{b1} + F_{b2})/m$. At the instant the robot decelerates, a plumb bob suspended from the center of gravity will be vertical. If the braking occurs too rapidly, as shown in Fig. 2(d), $F_2 = 0$, $F_1 = mg$, and the moment $F_{b1}h$ will be unopposed. When the plumb bob swings past the center of support at the forward wheel, the robot will begin to tip over. The situation is exacerbated by uneven floors and the fact that the effective robot radius is actually somewhat greater than r_1 to provide some margin of clearance when passing obstacles. Further, when the robot moves, its sensors (which must be located high on the body to interact with people) are subjected to a great deal of random motion because of the relatively small wheel bases, uneven floors, and “give” in the suspension. If the robot has a manipulator that can pick up and carry heavy objects the situation is even worse because the center of mass and inertial properties are rapidly changing [4].

In practice, to ameliorate these difficulties, these mobile robots have (1) bases that are relatively wide with respect to environmental dimensions, *e.g.*, doorways (increasing d), (2) have centers of gravity that are as low as possible (decreasing h , often achieved by the inclusion of significant dead weight), and (3) operate slowly enough to avoid tipping over (reducing F_{b1} and F_{b2}). This is not so bad for many purposes, but is it really the best we can do? It is tempting to dismiss these considerations as mere engineering details which are unimportant compared to the “real” classical problems dealing with perception, navigation, cognition, interaction, etc. It would be

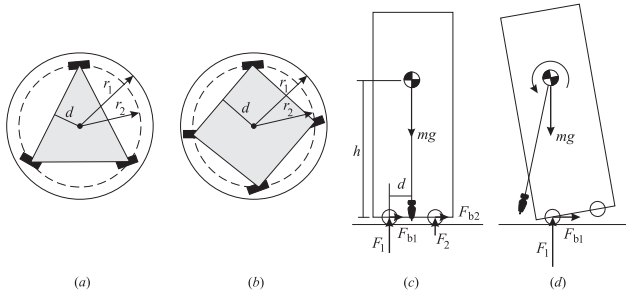


Fig. 2. Stability of conventional wheeled mobile robots: (a) three-wheeled base, (b) four-wheeled base, (c) stability margin, (d) tipping moment during acceleration or deceleration.

a mistake, however, to ignore the stability problem. Statically-stable wheeled mobile robots may be an evolutionary dead end when it comes to operating in human environments.

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.

II. BACKGROUND

A two-wheeled robot with inverse pendulum control developed in Japan was demonstrated in 1994 [6]. The two-wheeled design eliminated the need for a third castoring wheel. The same group introduced a one-wheel balancing robot [9]. 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 chairs¹ and balancing 2-wheel “Segway personal mobility devices”² have been introduced. The 2-wheel RMP robotic platforms [10] 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 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.

¹Independence Technology, <http://www.indetech.com>.

²Segway human transporter, <http://www.segway.com>.

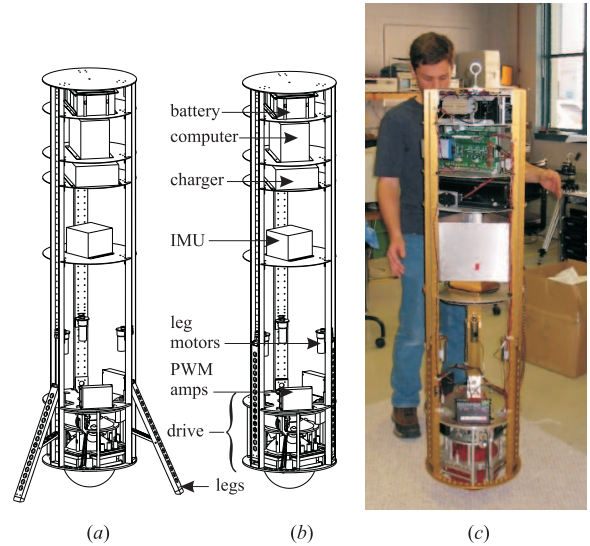


Fig. 3. Ballbot design and realization: (a) with three legs deployed, (b) with legs retracted into body, (c) balancing and station keeping.

III. SYSTEM DESCRIPTION

Ballbot, shown in Fig. 3, 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. Figs 3(a) and (b) show the design and Fig. 3(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 on board 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.

The drive mechanism, shown in Fig. 4, 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

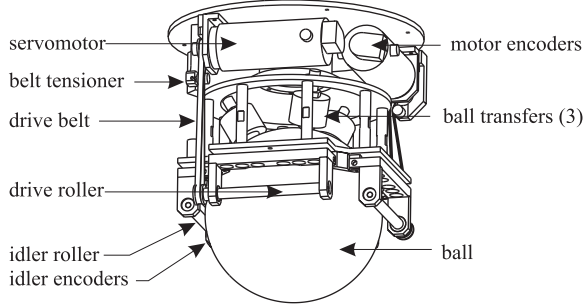


Fig. 4. Ballbot inverse mouse-ball drive mechanism

ball to produce motion. The initial ball was 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 the initial ball eventually wore out. A second ball design with a lighter 190.5 mm diameter spun aluminum shell and 12.7 mm of urethane has unobservable wear, presumably due to the lower shear stresses in the thicker urethane layer. 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.

IV. 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. Further, we assume that the motion in the median sagittal 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.

It is worth making special note of the modeling assumptions that are made regarding friction. Friction between the wheel and the floor and between the wheel and the body is modeled as pure viscous damping. Forces due to static friction and nonlinear dynamic friction are neglected. The inclusion of a viscous term in the friction model makes sense: there are

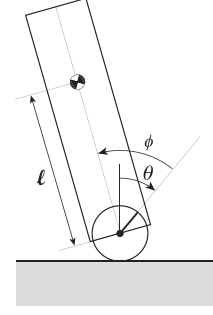


Fig. 5. Planar simplified Ballbot model used for controller design.

hysteresis losses associated with the compression and relaxation of the urethane layer that occurs at the ball-floor and ball-roller contact points, and these losses can reasonably be assumed to be velocity dependent. However the exclusion of the effects of static and nonlinear dynamic friction is not as easily justified, and we have determined experimentally that these effects are in fact significant (see Section VII). Still we choose to neglect these terms because they would introduce discontinuities and extreme nonlinearities that would render the resulting Ballbot model unusable for linear controller synthesis. As described in Section V, the controller presented here employs an inner PI loop to mitigate the effect of these modeling omissions.

Fig. 5 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.*, [3]). 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

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

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

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

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

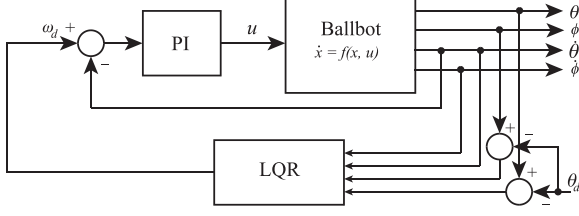


Fig. 6. Structure of stabilizing linear feedback controller.

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

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

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

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

V. 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. 6. 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 . This inner loop automatically compensates for the various frictional torques that

must be overcome to achieve velocity tracking, thus reducing the effect of the unmodeled static and dynamic friction. The integral term adds an extra state to the system. Define the augmented state vector $x_a = [x^T x_5^T]^T$. The closed loop equations of motion of the inner loop can then be written as

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

The outer loop is designed by linearizing the inner loop equations of motion and applying LQR (see e.g., [1]). 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:

$$\dot{x}_a = \underbrace{\frac{\partial f_a}{\partial x_a} \Big|_{x_a=0, \omega_d=0}}_A x_a + \underbrace{\frac{\partial f_a}{\partial \omega_d} \Big|_{x_a=0, \omega_d=0}}_B \omega_d.$$

The pair (A, B) is controllable, from which we can infer the absence of any nonholonomic constraints and the existence of a smooth stabilizing controller.

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

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. 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. 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. This may also be a consequence of the decision to neglect static and nonlinear dynamic friction in the simplified model.

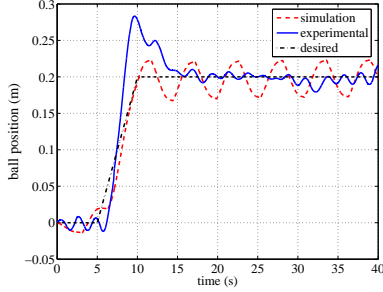


Fig. 7. Position step response for a point-to-point move.

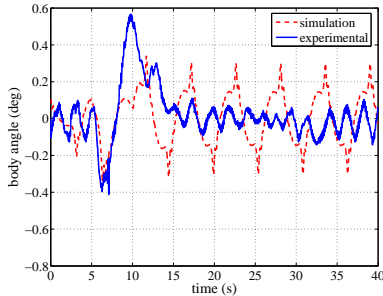


Fig. 8. Body angle during point-to-point move.

VI. 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 Figs. 7 and 8, 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. Differences between simulation and

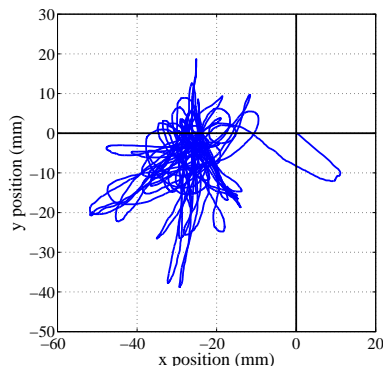


Fig. 9. Plot of the ball path during balancing and station keeping

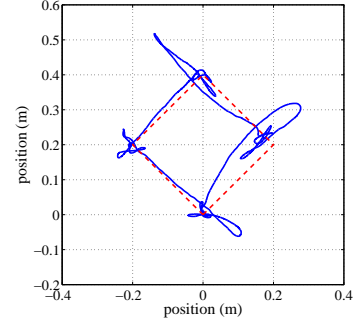


Fig. 10. Plot of the ball path while attempting to move in a square.

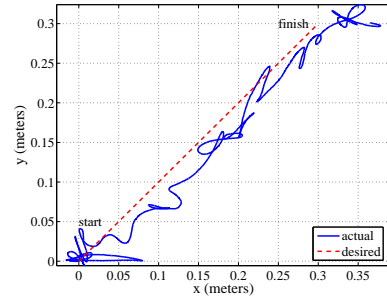


Fig. 11. Plot of the ball path for straight line move with trajectory control.

experiment are probably due to unmodeled frictional and spring forces. The divergence when station keeping is at most about 40 mm in position, and 0.5° in tilt.

To see the typical motion jitter experienced during operation, one may plot the paths taken as the ball moves around on the carpeted floor. Fig. 9 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. Fig. 10 shows Ballbot's attempt to track a square trajectory.

The straight line path plotted in Fig. 11 was generated by commanding Ballbot to move at a constant velocity for a period of 40 seconds, demonstrating that we can specify trajectories in terms of both desired position and desired velocity. This motion is much slower than the motion depicted in the previous straight line plot (Figs. 7 and 8) and in the square plot (Fig. 10). These fast motions exhibit fairly straight trajectories because Ballbot is essentially "falling" toward its goal, then recovering when it reaches the goal. In contrast, the motion depicted in Fig. 11 is slow, steady, and tightly controlled over the entire path.

VII. FUTURE WORK

The LQR controller presented here is sufficient to balance Ballbot and drive it along rudimentary trajectories on carpeted surfaces. These early capabilities fall short of the robust balancing and agile mobility that will be required in order for Ballbot to operate effectively in human environments.

The overly simplified friction model is likely to be a major reason for the poor performance of the resulting controller. Experimentally, we have determined that the effects of static

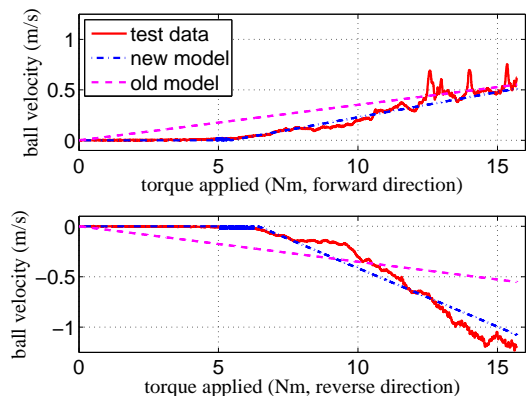


Fig. 12. Plot of ball velocity vs. applied torque. The plot compares experimental results to values predicted by the pure viscous model used in this paper and the nonlinear model that will be used in future work.

and nonlinear dynamic friction on Ballbot behavior are significant. For example, the break-away torque necessary to overcome static friction is nearly half of the maximum torque that can be applied by the motor. Additionally, the frictional effects are asymmetric; Ballbot moves in the reverse direction much more easily than it moves in the forward direction. We have developed a Ballbot actuation model that includes static, Coulomb, and viscous friction as well as the observed asymmetry. This improved model matches experimentally observed Ballbot motion significantly better than the pure viscous friction model used in this paper (see Fig. 12). In future work, we will use this model to guide the design of a more robust nonlinear balancing controller. Specifically, we will investigate the use of sliding mode control, which has been proven to be effective for systems with high static friction [5].

In addition to designing new controllers, we will make a number of mechanical modifications that will expand Ballbot's capabilities. One such modification is the installation of the actuator to control the yaw of the Ballbot body. This actuator controls the relative yaw angle between the body and the inverse mouse-ball actuator at its base, and it relies on static friction between the ball and the floor in order to control the absolute yaw of the body. For the aluminum ball with a half-inch thick coating of 70A-72A durometer urethane, we have measured the yaw-axis break away torque between the ball and the floor to be approximately 4.25 Nm on tile floor and 3.55 Nm on carpet. The area of contact between this ball and the floor is a circle with a diameter of 30mm. These values should be sufficient to prevent the ball from slipping during yaw maneuvers.

We will also consider a redesign of the inverse mouse-ball actuator. The observed asymmetry in the mouse-ball actuation is due to force imbalance that results from driving the ball from a single roller on the side. When the actuator moves forward, the ball is pushed up into the body, increasing the friction between the ball and body. When the actuator moves backward, the ball is pulled down, away from the body, decreasing friction. We will investigate designs that eliminate this asymmetry by

actuating the idler rollers so that the ball is driven from both sides with equal and opposite tangential forces.

VIII. 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, omni-directional 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.

Acknowledgment

This work was supported in part by NSF grant IIS-0308067. The authors also thank Eric Scheerer and Anish Mampetta for their indispensable assistance in running Ballbot experiments.

REFERENCES

- [1] B.D.O. Anderson and J.B. Moore. *Optimal Control: Linear Quadratic Methods*. Prentice Hall, 1990.
- [2] G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsch, D. Magaritis, M. Montemerlo, J. Pineau, N. Roy, J. Schulte, and S. Thrun. Towards personal service robots for the elderly. In *Proc. Workshop on Interactive Robotics and Entertainment (WIRE)*, Pittsburgh, PA, 2000.
- [3] A.M. Bloch. *Nonholonomic Mechanics and Control*. Springer, 2003.
- [4] A. Diaz-Calderon and A. Kelly. On-line stability margin and attitude estimation for dynamic articulating mobile robots. *Int'l. J. of Robotics Research*, 2005. (to be published).
- [5] C. Edwards and S.K. Spurgeon. *Sliding Mode Control: Theory and Applications*. Taylor and Francis Ltd, 1998.
- [6] 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.
- [7] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, and A. Casal. Vehicle/arm coordination and multiple mobile manipulator decentralized cooperation. In *Proc. of the IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pages 546–553, Osaka, 1996.
- [8] T. Lauwers, G. Kantor, and R. Hollis. One is enough! In *Proc. Int'l. Symp. for Robotics Research*, San Francisco, October 12-15 2005. Int'l. Foundation for Robotics Research.
- [9] 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.
- [10] 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.
- [11] R. Simmons, J. Fernandez, R. Goodwin, S. Koenig, and Joseph O'Sullivan. Xavier: An autonomous mobile robot on the web. *Robotics and Automation Magazine*, 1999.
- [12] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, Frank Dellaert, Dieter Fox, D. Haehnel, Chuck Rosenberg, Nicholas Roy, Jamieson Schulte, and D. Schulz. MINERVA: A second generation mobile tour-guide robot. In *Proc. of the IEEE Int'l Conf. on Robotics and Automation (ICRA '99)*, 1999.