

Initial Results for a Ballbot Driven with a Spherical Induction Motor

Greg Seyfarth¹, Ankit Bhatia¹, Olaf Sassnick², Michael Shomin¹, Masaaki Kumagai³, and Ralph Hollis¹

Abstract—We give a system overview and initial experimental results for SIMbot (spherical induction motor based ballbot), a human-sized dynamically stable mobile robot with only two active moving parts. SIMbot is a qualitatively new kind of robot with the potential to drastically reduce mechanical complexity in human-sized mobile robots. SIMbot uses a novel spherical induction motor as its drive mechanism, which is mechanically simpler than other ballbot drive mechanisms. We demonstrate the feasibility of this innovation with experimental results for balancing, stationkeeping, short point-to-point motions, and recovery from initial lean angles.

I. INTRODUCTION

As interest in mobile robots for domestic and service applications grows, efficient, graceful, and compliant locomotion for mobile robots remains an open question. Ballbots, human-sized dynamically stable mobile robots that balance on a single spherical wheel, were first successfully developed in 2005 [1] [2] to provide locomotion capabilities superior to those of statically stable wheeled mobile robots.

Ballbots are better suited to navigating in cluttered human environments than their statically stable, multi-wheeled counterparts. Traditional statically stable mobile robots must have a wide base and a low center of gravity for stability, resulting in a large footprint and difficulty in navigating through tight spaces. Their acceleration capabilities are limited by their stability requirements, and as a result, statically stable mobile robots are traditionally “fat” and slow. Ballbots offer several advantages over traditional statically stable mobile robots, including omnidirectionality, physical compliance, and the ability to use their weight to help with heavy loads [3], all while allowing a high center of mass (COM) and a human form factor.

Many drive mechanisms for ballbots have been proposed, each with various advantages and disadvantages. The original ballbot developed at Carnegie Mellon University relies on an inverse mouse-ball (IMB) drive to actuate the urethane covered ball, which works by squeezing the ball with four metal rollers attached to high-torque dc motors with timing belts. BallIP [4] and Rezero [5] use omniwheel-based drive systems. Both the IMB drive and the omniwheel-based drives are mechanically complex. The IMB drive suffers from

*This work was supported in part by NSF grant ECCS-1102147. Sassnick was supported by a Marshall Foundation Fellowship.

¹Greg Seyfarth, Ankit Bhatia, Michael Shomin and Ralph Hollis are associated with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213, USA. (gseyfart, ankitb, mshomin)@andrew.cmu.edu, rhollis@cs.cmu.edu

²Olaf Sassnick is a Masters candidate at the Salzburg University of Applied Sciences, Salzburg, Austria. osassnick.its-m2013@fh-salzburg.ac.at

³Masaaki Kumagai is an Associate Professor in the Faculty of Engineering, Tohoku Gakuin University, Tagajo 985-8537 Japan. kumagai@tjcc.tohoku-gakuin.ac.jp



Fig. 1: SIMbot balancing next to a human for size comparison.

excessive friction because the rollers parallel to the direction of motion of the ball must slip along the ball’s surface. It also has a complex transmission with several potential failure points. The timing belts, for instance, can wear, and require maintenance. Furthermore, the belts require precise tensioning, and the performance of the IMB is sensitive to the amount of pressure exerted on the ball by the rollers. Both of these factors result in a drive system which can be time-intensive to tune and to fix. In omniwheel-based drive systems, the omniwheels support the entire weight of the robot. As a result, the omniwheels must be of a certain minimum size to have the strength to support the weight of the robot. Because of this size requirement, it is generally necessary to introduce gears into the transmission in order to generate enough torque on the ball. Gears introduce additional points of failure, friction, and backlash into the system. Omniwheels are also very mechanically complex and have many small parts.

A continuous rotation spherical motor would greatly simplify the drive mechanism. Although a number of spherical motors have been developed previously [6] [7] [8], typically relying on electromagnetic or ultrasonic actuation, none have the speed and torque characteristics necessary to drive a ballbot. In light of this problem, we previously developed a spherical induction motor [9], or SIM, specifically to have

the performance necessary to drive a ballbot.

In this paper, we present SIMbot, a ballbot which uses a SIM as a drive mechanism. SIMbot has only two active moving parts—the body and the ball. We see SIMbot as having two primary advantages over other drive systems. Firstly, by virtue of its simplicity, SIMbot comes very close to an ideal ballbot from a modeling perspective by removing many of the unwanted and unmodeled dynamics such as gear friction and backlash and timing belt slop in the transmission of previous drive systems. Secondly, SIMbot is extremely simple mechanically: the only mechanical parts of the SIM drive system which require tuning or maintenance are the ball transfers which support the body. We project that SIMbot will possess all the traditional benefits of mechanical simplicity, including reliability, easy serviceability, and possibly lower cost. The SIM also has much less friction than the IMB drive, which could improve efficiency. We expect SIMbot to be a key technology in making ballbots more accessible and practical for wide adoption.

This paper focuses primarily on the system of and experimental results for SIMbot rather than the control and modeling scheme of the SIM. For more information on the SIM, refer to [9]. For more information on our dynamics model and control scheme, refer to [10].

II. SYSTEM OVERVIEW

A. Spherical induction motor

The spherical induction motor, shown in Fig. 2, consists of a 7075-T6 aluminum alloy frame with six laminated stators mounted around the ball, or rotor, at a nominal 1 mm air gap. The stators have twelve slots and are wound with nine coils in a three-phase, four-pole configuration. Each coil consists of twenty-five turns of AWG 19 double polyimide insulated wire. The rotor is a hollow soft steel core electroplated with solder and joined to an outer copper shell via shrink-fitting. Eddy currents in the copper shell provide reaction torques on the rotor.

The force on the rotor due to each individual stator can be approximated as acting at a single point. We assume that this point is at the center of the stator and that the force acts in a direction parallel to the stator. In order to allow for yaw motion of the rotor, the stators are skewed from the rotor's polar axis by 10° .

Six nylon ball transfers support the entire weight of the robot over the ball and maintain the air gap. Nylon ball transfers were used rather than steel ball transfers to protect the soft copper shell of the ball. Since the ball transfers are the only point of contact between the ball and the body, SIMbot benefits from much lower friction than the IMB drive, which requires the rollers to tightly squeeze the ball to ensure traction.

To obtain closed loop torque control of the rotor, closed loop force control is performed on each individual stator. Each stator uses a field-oriented controller (the vector drives in Fig. 2) which closes the loop on stator current at 10 KHz. For more detail on this controller, see Section III. An actuator matrix $M_a \in \mathbb{R}^{3 \times 6}$ can be defined which maps the forces

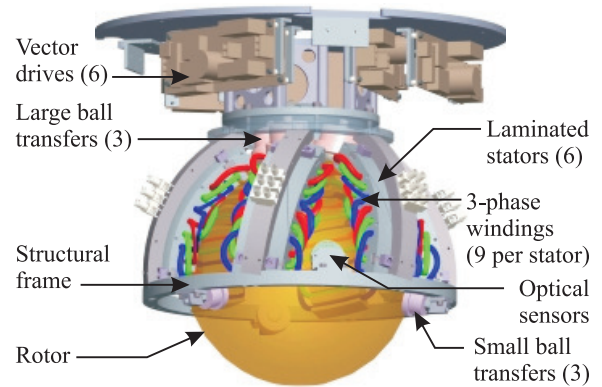


Fig. 2: A CAD model of the six-stator spherical induction motor used by SIMbot.

F_i , $i = 1, \dots, 6$, applied by the stators to the torque $\tau \in \mathbb{R}^{3 \times 1}$ about the center of the rotor:

$$M_a F = \tau. \quad (1)$$

Since the motor is over-actuated, having six applied forces for only three degrees of freedom, we are not limited to a single set of forces for a desired torque. The pseudo-inverse $M_a^+ = (M_a^T M_a)^{-1} M_a^T$ can then be used to solve for F_d , the minimum L_2 -norm vector that produces a desired torque τ_d .

With the existing vector drivers, each individual stator is capable of continuously producing tangential forces of up to 40 N and up to 70 N for up to 0.5 seconds. The motor was tested at net torques of up to 8 Nm with all stators active. These torques are presently limited by the electronics of the vector drivers.

For a detailed derivation of M_a and further details on the motor's design, construction, and performance characteristics, refer to [9].

B. Sensing

SIMbot uses a VectorNav™ VN-100 inertial measurement unit (IMU) for sensing body angles and body angular rates. The VN-100 has an onboard Kalman filter which provides estimates of the sensor attitude.

SIMbot relies on a set of three Avago™ ADNS-9800 optical gaming mouse sensors, shown in Fig. 2, positioned at 120° increments along the equator of the ball to sense the ball's angular velocity. The mouse sensors report the number of counts of displacement at 134 counts per millimeter by tracking the surface texture of the ball's surface. Six surface velocities are estimated, two from each mouse sensor, with a simple finite difference approximation. A least-squares scheme is used to infer the ball's angular velocity from these six estimated surface velocities. This sensing scheme is described in more detail in [9].

C. Communication and processing

SIMbot uses an Intel™ NUC 5i5MYHE running the QNX real-time operating system to perform all balancing

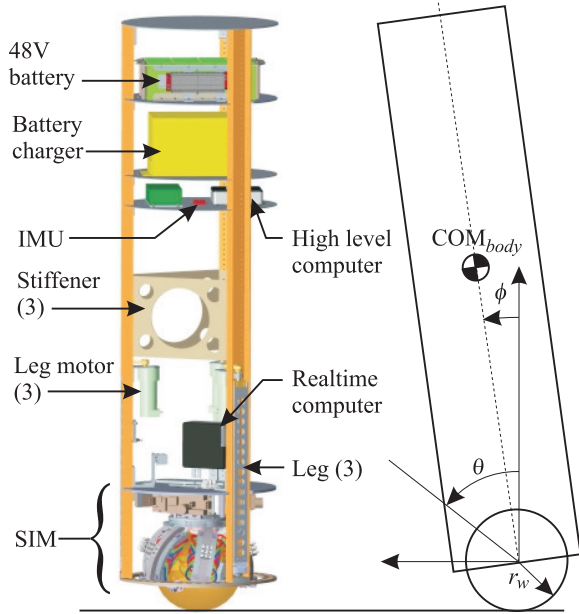


Fig. 3: Left: An approximate CAD model of the entire robot, showing the relative location of key components. Right: Planar model for a ballbot.

TABLE I: Physical system parameters

Parameter name	Symbol	Value
Mass of the body	m_b	41.98 kg
Mass of the ball	m_w	7.45 kg
Moment of inertia of the ball	I_w	0.05 kg · m ²
Ball radius	r_w	0.1013 ± 0.00012 m
COM of body above center of ball	l_b	0.679 m
Roll moment of inertia of the body ¹	I_b^{xx}	10.343 kg · m ²
Pitch moment of inertia of the body ¹	I_b^{yy}	10.385 kg · m ²
Yaw moment of inertia of the body ¹	I_b^{zz}	0.485 kg · m ²

¹Estimated from CAD model

calculations. This real-time computer communicates with the VectorNav™ VN-100 IMU via serial, the leg motor drivers via an I²C bus, and the SIM vector drivers and “mouse interface” via a second I²C bus. The mouse interface is an intermediate PIC microcontroller which reads the three individual mouse sensors via an SPI bus. We convert the real-time computer’s native USB to I²C and serial with Devantech USB-ISS modules. Fig. 4 shows a block diagram of the entire communication scheme.

The real-time computer closes the balance controller loop at 200 Hz, currently limited by overhead introduced by converting from USB to I²C. It is expected that the control loop rate can be improved by switching to native I²C communication.

An Intel™ NUC D34010WYK performs trajectory generation. This computer will be used for other high-level computations such as navigation and perception in the future. It runs Robot Operating System (ROS) on Ubuntu Linux, giving access to many existing packages for navigation and mapping.

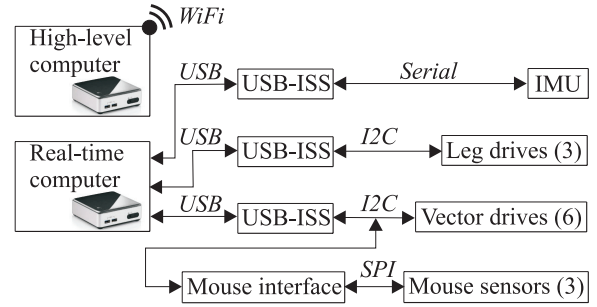


Fig. 4: System architecture for SIMbot.

D. Legs

SIMbot has three legs which are conceptually identical to those used on the IMB-drive ballbot. The legs are used only for stability when SIMbot is powered down. Each leg is driven by a linear drive screw powered by a dc motor with an attached quadrature encoder. The end of each leg has a spring-loaded limit switch with a ball caster. The limit switch detects contact with the floor and with the body when the legs are going down and up, respectively. Each leg is run by a separate motor driver with an integrated PID controller for velocity control of each individual leg.

III. DYNAMICS AND CONTROL SCHEME

A. Dynamics

1) *Modeling assumptions:* Following the approach of the IMB-based ballbot, the dynamics of SIMbot are simplified to two linearized planar models, one in the median sagittal plane and one in the median coronal plane. The model is based on several assumptions:

- 1) The motion in the median and sagittal plane is decoupled
- 2) The motion in each of these two planes is identical
- 3) The ball rolls without slipping
- 4) The floor is flat and level

These assumptions are reasonable as the coupling between sagittal and coronal planes is through products of body angular rates and body lean angles, both of which should be very small. As a result, the linearized 3D model is, in fact, decoupled in the sagittal and coronal plane. Using the two planar models, we design two identical stabilizing PID controllers.

2) *Planar model:* The SIMbot in each orthogonal plane is abstracted to a rigid rectangle on top of a rigid disk. As shown in Fig. 3, state variables θ and ϕ define the ball angle and body lean angle respectively. The equations of motion for the planar ballbot are as follows:

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) + D(q, \dot{q}) = \begin{bmatrix} \tau \\ 0 \end{bmatrix}, \quad (2)$$

where $q = [\theta, \phi]^T$ is the generalized coordinate vector, $M(q)$ is the inertia matrix, $C(q, \dot{q})$ is the Coriolis matrix, $G(q)$ defines the gravitational forces, $D(q, \dot{q})$ are an approximation of the frictional forces acting on the ball and τ is the applied torque between body and ball. It is interesting to

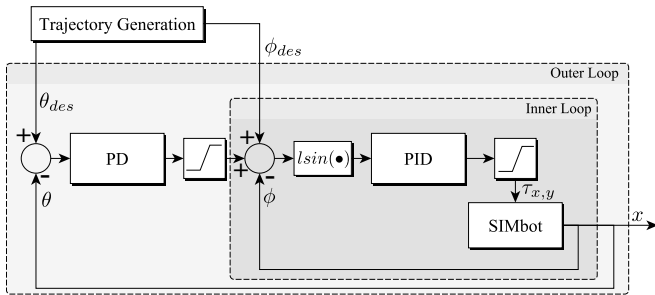


Fig. 5: Block diagram of the balancing and outer loop controllers. Note: the $lsin(\bullet)$ block maps the lean angle error to a displacement of the center of mass from the vertical.

note that although the SIM itself is overactuated, SIMbot is an underactuated system, having no actuator which directly controls its body lean angle.

B. Control Architecture

The SIMbot control scheme is based on three nested controllers. The controller at the lowest level is encapsulated by the balancing controller and finally the outer loop controller. A nested yaw controller is responsible for achieving the desired yaw angles. Fig. 5 shows the balancing controller nested inside the outer loop controller and Fig. 6 details the yaw controller.

1) *Vector Controller*: The lowest level controller is a field-oriented vector control scheme: one for each of the six stators in the SIM. The vector controller takes as input the force command for a stator and generates a 3-phase ac voltage to produce that force. More information about the vector control scheme can be found in [11].

2) *Balancing Controller*: The balancing controller maintains a given lean angle setpoint. This is a conventional hand tuned PID controller identical to the one used on the IMB-drive ballbot [10]. Two instances of this controller handle balancing in each of the orthogonal directions. The balancing controller determines the torque τ to be generated by the SIM in order to maintain the lean angle setpoint ϕ_{des} . A saturation on output torque prevents the controller from commanding excessive torque from the SIM.

3) *Yaw Controller*: In order to command a desired yaw azimuth of the robot, a nested PID controller commands yaw torques through the actuator matrix. The controller (see Fig. 6) consists of an inner loop PI controller that closes the loop on yaw angular velocity and an outer loop PD controller feeding back yaw angle (integrated from yaw angular velocity reported by the IMU) and velocity. The output of both controllers is saturated to prevent high yaw torques.

4) *Outer Loop Controller*: The outer loop controller tracks the desired ball position by outputting desired body lean angles. The error between desired ball position θ_{des} and the observed ball position θ is driven to zero using the PD controller. The outer loop controller can act as a station-keeping controller which maintains ball position or as a trajectory tracking controller for point-to-point motions.

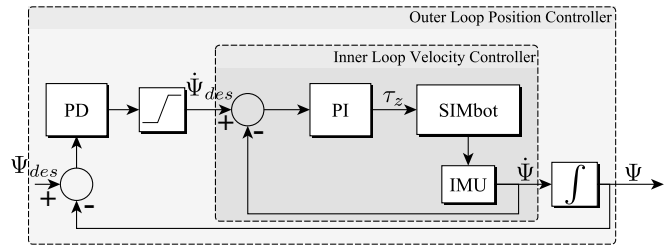


Fig. 6: Yaw controller: an inner velocity controller and outer loop position controller achieve a desired yaw angle.

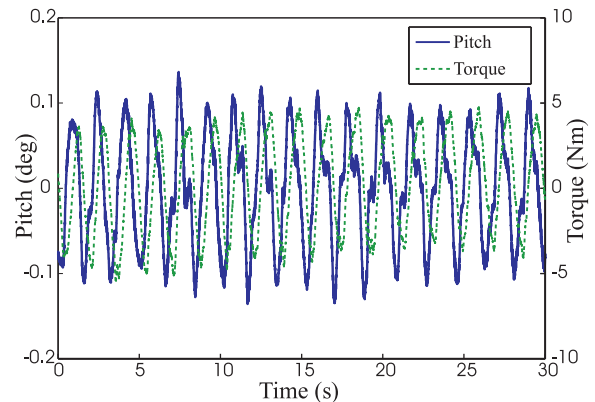


Fig. 7: SIMbot attempting to maintain a zero lean angle over thirty seconds on a soft foam tile. Similar results were obtained in the other plane.

Motion planning for ballbots is not a trivial problem, and a number of planners have been proposed, e.g., [12]. In this work, we generate trajectories with the differential flatness based motion planning scheme given in [13].

IV. EXPERIMENTAL RESULTS

We performed four experiments to test our initial controller design and hardware capabilities.¹ We examined balancing performance while maintaining a zero lean angle, stationkeeping performance, performance for a point-to-point motion, and recovery from an initial lean angle. For all experiments, position data was obtained from the mouse sensors on the SIM and orientation data was obtained using the IMU on SIMbot. All experiments were performed on 9.5 mm thick soft foam tiles to protect the soft copper layer of the ball. The tiles add additional rolling resistance which makes it easier for the robot to balance, since the tiles damp the motion of the ball. We will investigate a number of methods for protecting the soft copper shell to allow for motion on harder surfaces.

A. Maintaining zero lean angle

See Fig. 7 for the pitch angle and torque used to control the pitch angle during thirty seconds of SIMbot attempting to maintain a zero lean angle. This experiment was free from disturbances. SIMbot is able to maintain a zero degree lean angle to within $\pm 0.15^\circ$. Fig. 8 shows SIMbot's position over

¹Many of these behaviors can be seen in the video attachment, which is also available at <http://www.msl.ri.cmu.edu/projects/ballbot/simbot.html>.

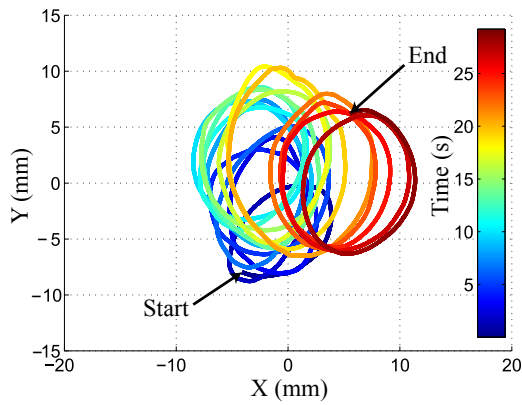


Fig. 8: A parametric plot of the position of SIMbot over the same 30 s period as in Fig. 7. SIMbot is not explicitly attempting to maintain a set position on the floor.

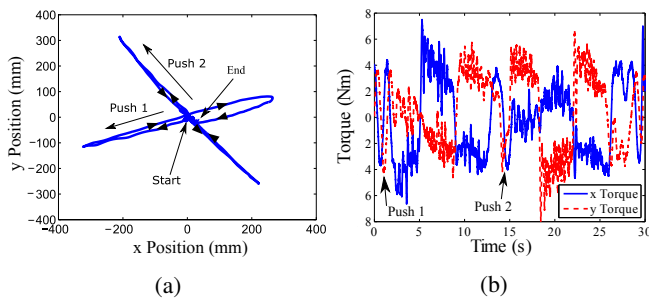


Fig. 9: Two pushes of roughly 10 N and lasting roughly 2 s were applied to SIMbot to displace it from its initial position. The stationkeeping controller brings SIMbot back to its initial position. Best viewed in color.

the same 30 s. Despite not explicitly controlling for position, SIMbot stays within 15 mm of its starting position during the entire balancing operation.

B. Stationkeeping

We applied roughly 10 N of force to the robot for about 2 s, once in each of two orthogonal directions, while running the stationkeeping controller. After undergoing initial displacements of up to 0.4 m, SIMbot is able to return to its initial position. See Fig. 9 for a plot of SIMbot’s position and the torques applied to the ball during the experiment.

C. Point-to-point motion

A 0.7 m point-to-point motion was demonstrated with SIMbot using the outer loop controller described in Section III-B to generate and track desired ball trajectories. See Fig. 10 for body angle tracking results and Fig. 11 for position tracking results.

D. Initial lean angle recovery

We tested SIMbot’s ability to return to a zero degree lean angle when starting from an initial non-zero lean angle. One experimenter held the robot stationary while another commanded the desired initial lean angle to SIMbot’s outer loop controller. SIMbot was then released and a recovery

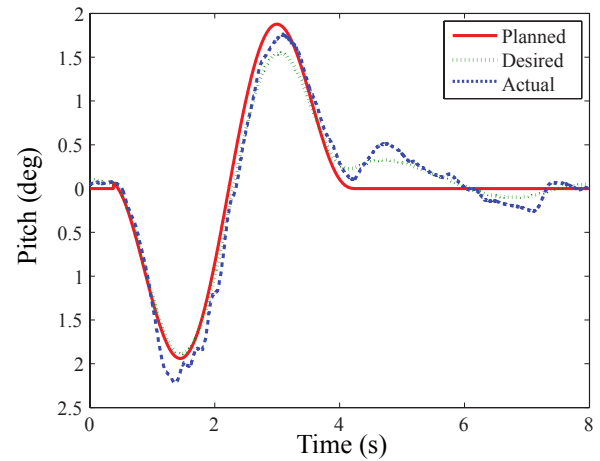


Fig. 10: SIMbot executing a 0.7 m point-to-point motion. “Planned” is the feedforward body angle trajectory output by our planner. “Desired” is the feedforward body angle trajectory combined with a PD controller on ball position. “Actual” is the trajectory followed by SIMbot.

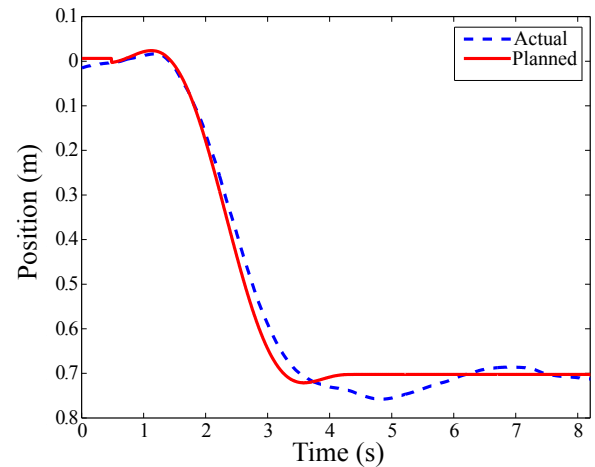


Fig. 11: The x position of SIMbot during a 0.7 m point-to-point motion.

trajectory was planned and executed. The recovery trajectory is designed to take SIMbot from its initial state to a final resting state with zero lean angle and zero ball velocity. As we are primarily concerned with returning the lean angle to zero degrees in this experiment, we did not include the ball position feedback term in the outer loop controller. SIMbot was able to recover from initial lean angles of up to 3° as shown in Fig. 12.

V. DISCUSSION

When attempting to maintain a zero lean angle, SIMbot’s performance is comparable to the IMB-based ballbot in both “zero-point motion” (the amount of position drift while not explicitly attempting to maintain position) and lean angle error about zero degrees. SIMbot’s performance in behaviors requiring more torque such as more aggressive point-to-point motions and recovery from larger lean angles is currently limited by our vector drivers, which can produce torques of up to approximately 8 Nm. We are in the process of

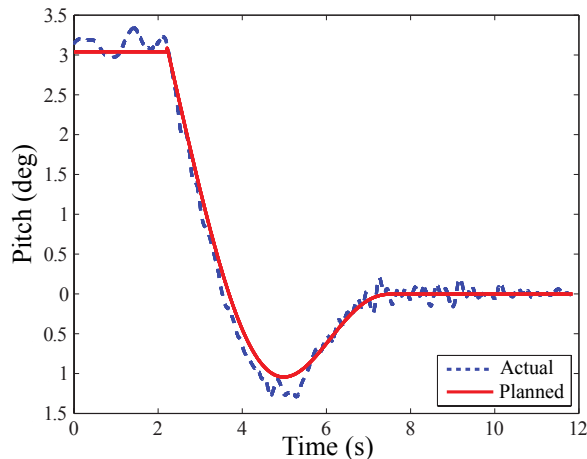


Fig. 12: SIMbot recovering from an initial lean angle of roughly 3° in the y-direction.

developing more rugged motor drivers which should improve the maximum torque by a factor of 3. It is encouraging that the pure balancing motion, stationkeeping, and the short trajectory with a maximum lean angle of less than 2.5° are all feasible with our 8 Nm limit. For comparison, the IMB-based ballbot operates in a nominal body angle range of $\pm 5^\circ$ and demonstrated a 12° body lean angle in [3]. A simple simulation in [9] which did not take into account the rolling resistance from the foam tiles showed SIMbot recovering from a lean angle of 5° with the output torque saturating at 8 Nm for nearly a second. In practice, however, SIMbot was not able to generate enough torque to recover from lean angles greater than 3° , and this is likely because the simulation did not include the additional rolling resistance added by the foam tiles and used a very simple friction model for the SIM.

From a modeling perspective, SIMbot is a much closer approximation to the idealized ballbot model: an omnidirectional two-rigid-body robot. SIMbot eliminates many of the unmodeled dynamics in the IMB drive such as timing belt slack and also helps to reduce the effect of unintended nonlinearities such as stiction between the body and the ball. In future work, the ball transfers between the ball and the body could be replaced with an air bearing. This would eliminate nearly all friction between the body and the ball and result in the idealized ballbot model being an extremely high fidelity approximation of SIMbot. We expect that SIMbot will be an excellent testbed for ballbot controllers by leveraging its faithfulness to a simple, two-rigid-body model of a ballbot.

VI. CONCLUSION

We have demonstrated that it is possible to use a spherical induction motor as a primary drive mechanism for a ballbot. This drive mechanism reduces the entire ballbot to two active moving parts—the body and the ball—and greatly reduces the mechanical complexity of ballbots when compared with previous designs. As a consequence of its mechanical simplicity, SIMbot comes very close to an ideal ballbot consisting of only two rigid bodies. We expect that

this development will be a major enabler in moving ballbots from research labs to real world applications.

VII. FUTURE WORK

Much work remains to be done on SIMbot. In particular, we would like to compare SIMbot to the IMB-based ballbot directly on metrics such as energy expenditure while balancing, energy expenditure while accelerating and executing point-to-point motions, maximum acceleration, and maximum speed. To perform this comparison, more powerful vector drivers will be needed to allow the SIM to generate larger torques, and additional mass will be added to SIMbot to keep the inertial properties of the two robots similar. In informal experiments, we observed SIMbot drawing around 20 A (at 48 volts) while balancing, and we expect the primary limitation of the SIM to be its maximum torque output rather than its top speed.

There are many avenues for optimizing the design of SIMbot. Many parts of the SIM itself could be optimized, including adjusting the wire gauge for either higher performance or longer battery life, optimizing the thickness of the ferromagnetic core and the soft copper shell on the ball, and designing longer stators to minimize the end-effects of the magnetic fields.

REFERENCES

- [1] T. Lauwers, G. Kantor, and R. Hollis, “One is enough!” in *Proc. Intl. Symp. for Robotics Research*, 2005, pp. 12–15.
- [2] —, “A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive,” in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 2884–2889.
- [3] M. Shomin, J. Forlizzi, and R. Hollis, “Sit-to-stand assistance with a balancing mobile robot,” in *Proceedings IEEE International Conference on Robotics and Automation*, 2015.
- [4] M. Kumagai and T. Ochiai, “Development of a robot balancing on a ball,” in *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*. IEEE, 2008, pp. 433–438.
- [5] S. Doessegger, P. Fankhauser, C. Gwerder, J. Huessy, J. Kaeser, T. Kammermann, L. Limacher, and M. Neunert, “Rezero,” *Focus Project Report, Autonomous Systems Lab., ETH Zurich, Switzerland*, 2010.
- [6] S. Toyama, S. Sugitani, Z. Guoqiang, Y. Miyatani, and K. Nakamura, “Multi degree of freedom spherical ultrasonic motor,” in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 3. IEEE, 1995, pp. 2935–2940.
- [7] K.-M. Lee, H. Son, J. Joni, *et al.*, “Concept development and design of a spherical wheel motor (SWM),” in *IEEE International Conference on Robotics and Automation*, vol. 4. IEEE; 1999, 2005, p. 3652.
- [8] K. Kaneko, I. Yamada, and K. Itao, “A spherical dc servo motor with three degrees of freedom,” *Journal of dynamic systems, measurement, and control*, vol. 111, no. 3, pp. 398–402, 1989.
- [9] A. Bhatia, M. Kumagai, and R. Hollis, “Six-stator spherical induction motor for balancing mobile robots,” in *Proceedings 2015 IEEE International Conference on Robotics and Automation*. IEEE, 2015, pp. 226 – 231.
- [10] U. Nagarajan, G. Kantor, and R. Hollis, “The ballbot: An omnidirectional balancing mobile robot,” *The International Journal of Robotics Research*, vol. 33, no. 6, pp. 917–930, May 2014.
- [11] M. Kumagai, “Development of a linear induction motor and a vector control driver,” in *SICE Tohoku chapter workshop material*, 2010, pp. 262–9.
- [12] U. Nagarajan and R. Hollis, “Shape space planner for shape-accelerated balancing mobile robots,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1323–1341, 2013.
- [13] M. Shomin and R. Hollis, “Differentially flat trajectory generation for a dynamically stable mobile robot,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4467–4472.