

# Robust Balancing and Position Control of a Single Spherical Wheeled Mobile Platform

Firat Yavuz and Mustafa Unel

Faculty of Engineering and Natural Sciences  
Sabanci University, Tuzla, Istanbul 34956, Turkey  
{firatyavuz, munel}@sabanciuniv.edu

**Abstract**—Self-balancing mobile platforms with single spherical wheel, generally called ballbots, are suitable example of underactuated systems. Balancing control of a ballbot platform, which aims to maintain the upright orientation by rejecting external disturbances, is important during station keeping or trajectory tracking. In this paper, acceleration based balancing and position control of a single spherical wheeled mobile platform that has three single-row omniwheel drive mechanism is examined. Robustness of the balancing controller is achieved by employing cascaded position, velocity and current control loops enhanced with acceleration feedback (AFB) to provide higher stiffness to the platform. The effectiveness of the proposed balancing controller is compared with commonly used optimal state feedback method. Additionally, the position controller is designed by utilizing the dynamic conversion of desired torques on the ball that are calculated from virtual control inputs generated in the inertial coordinates. Dynamical model of a ballbot platform is investigated by considering highly nonlinear couplings. Performance of the controllers are presented via simulation results where the external torques were applied on the body in order to test disturbance rejection capabilities.

## I. INTRODUCTION

Research on omnidirectional and self-balancing mobile platforms has been quite active in robotics and control communities in the last decade due to their ability to freely move in all directions on the horizontal plane. There are two main types of omnidirectional robots, the conventional wheeled structure and the special wheeled structure. Self-balancing mobile platforms with single spherical wheel, also known as ballbot platforms, are suitable example of special wheeled systems. In such systems, a body balances on a single ball which used as a replacement of conventional wheels to achieve omnidirectional motion. The spherical wheel is mostly manipulated through rollers or wheels attached to the actuators on the body. In summary, single spherical wheeled mobile platforms must actively balance on a single point of contact with the ground which reduces energy for motion due to less friction but makes system to be inherently unstable. Beside being a research valuable topic, single spherical wheeled platforms are aimed to perform missions as accompanying handicapped or elder people to move around (e.g. in hospitals, museums, supermarkets), furthermore as a vehicle for personal transportation [1].

Single spherical wheeled platforms have been popular over the years, and many researchers have already paid attention on modeling, design and construction of such systems. The first ball balancing robot called ERROSphere (Equilibrating Robot Rolling On Sphere) [2], was presented by Havasi in

2005, merely can balance on the ball without any further functionality. B.B.Rider [1] was another omnidirectional personal transportation vehicle stabilizes on a basketball developed by Endo et al. Initial implementation on ballbot type robots, actually an inverted pendulum mounted on a spherical wheel, started in 2006. The first example of this morphology constructed by Lauwers et. al [3] as human size ballbot which has an inverse mouse-ball drive mechanism to actuate the ball and can interact with humans. Although the robot with the inverse mouse-ball driven mechanism worked well, it could not rotate around yaw axis. During the following years, many other version of the inverse mouse-ball driven ballbots were developed by adding yaw control and different features such as stabilizer legs [4] and functional arms [5]. Later on, many other configurations are developed by using different actuating systems. Most commonly, omniwheels driven by motors are used in many applications for better control performance [6], [7], where the Rezero [8] is the most famous one, to the best of our knowledge. Moreover, novel actuating mechanisms are presented in [9] as partially sliding rollers and in [10] spherical induction motor ball wheel designed exclusively for a ball balancing mobile robots. Additionally, several commercial ventures were exhibited from research to commercialization with the Rezero [8] for omniwheel structure and the mObi [11] for inverse mouse-ball drive structure.

Although much progress has been made on the development on spherical wheeled mobile platforms, controlling motion while maintaining balance is a challenging task because their underactuated structure has fewer control inputs than their degrees of freedom. In the literature, various balancing and positioning control algorithms for ballbot platforms were presented. The most common approach for controlling both the orientation and the position of the ballbot is LQR (Linear-quadratic regulator) based state feedback control. In [3], the ballbot controller consisted of an inner PI (Proportional+Integral) control loop for controlling angular velocity of the ball and an outer LQR based full state feedback control loop. Later on, in [4], [12] these controllers were replaced by PID (Proportional+Integral+Derivative) controller for both balancing and tracking set points. In [6], inputs of the plant were decoupled so that simple linear PD controllers were used to control virtual wheels. Then virtual accelerations were converted into the velocities of three omniwheeled stepping motors by using kinematic relations. Unfortunately neither LQR nor PID controllers was not sufficiently robust for balancing purpose to handle sudden deviations due to external forces and highly inclined initial conditions. Differently, Rezero is

controlled by quasi-nonlinear feed-forward control using the idea of gain scheduling [8]. In [13] inverse dynamics controller is used to cancel out accelerations on body attitude degrees of freedom, in combination with a PD controller to stabilize the body angles and the position. Another important issue is the path following control for the ballbot, which ensures that the robot to follow a path while maintaining balance. Initial research in position control of the ballbot focused on station keeping where the postural stabilization by an outer loop using LQR [3] or PID [4]. These approaches were successful in realizing station keeping and slow line following. In [5], [12] dynamic constraint-based optimal shape planner is used to plan shape trajectories which are used as reference for the body angles during tracking for the inverse mouse-ball driven ballbot.

In the aforementioned methods, LQR and PID controllers can also get the optimal results but were not sufficiently robust for balancing purpose to handle sudden deviations due to the external disturbances or highly inclined initial conditions. On the other hand using nonlinear controller is quite complicated because of the complex structure of the dynamic equation of the ballbot. Therefore, the development of robust approaches for controlling the ballbot type platforms is important. The robust stabilization problem is generally solved by using acceleration feedback (AFB) signals. Schmidt and Lorenz [14] have showed that acceleration feedback acts as an electronic inertia to provide higher stiffness to the platform. Thus, the overall system exhibits better disturbance rejection.

In this paper, angular acceleration signals are incorporated as feedback to the cascaded position, velocity and current control loops which are used to provide robust performance against the disturbing influences on the ballbot. Proposed method is compared with the commonly used idea of state feedback control which in this work is extended to a convex optimization problem by applying the efficient tools of Linear Matrix Inequalities (LMI) [15]. The primary objective of acceleration feedback control is rejecting disturbance which manifests itself directly in the acceleration signal. On the other hand, the goal of LMI is to find feedback gain matrix  $K$  by updating parameters of the control law according to the Lyapunov Stability Theorem. Also, we address the question of locomotion by using the position controller based on virtual control inputs generated in the inertial coordinates to find desired angular acceleration of the ball, then motor torques are calculated by utilizing the dynamic conversion. Initially, 3D dynamic model of a ballbot platform is derived by considering the highly nonlinear couplings and uncertainties in order to implement proposed control algorithms. Results demonstrate better performance of disturbance rejection for the cascaded controller enhanced with acceleration feedback over the state feedback controller which does not utilize such feedback.

The remainder of this paper is organized as follows. Section II details the dynamic model of a ballbot. Section III explains the fundamental design of the acceleration based cascaded controller, LMI based controller and the position controller schemes. In Section IV, the effectiveness of the proposed control algorithms are validated by numerical simulations. Finally, conclusions and directions for future work given in Section V.

## II. SYSTEM DESCRIPTION

Generally a ballbot consists of two main components: a spherical ball and an upper body placed on top of it. The model used in this work is three omnivheeled morphology where the body includes three single-row omnivheel driven by three independent dc motors which are placed with  $\beta$  degree between each other at the bottom. It is assumed that an omnivheel can only apply forces to the ball in the tangential direction of its rotation in the body reference system. Omnivheels are wheels with small rollers around the periphery which are perpendicular to the turning direction. The only interaction between ball and body is exist through those omnivheels as shown in Figure 1,

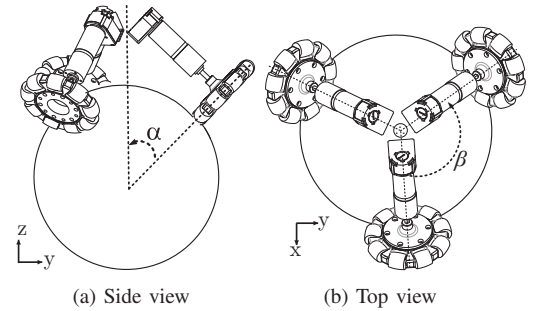


Fig. 1: Ballbot omnivheel configuration.

where  $\alpha$  and  $\beta$  are the angles of the ball-motor configuration.

The 3D ballbot system is defined in the inertial coordinate frame as shown in Figure 2. The origin of the coordinate frame used to derive the dynamic model is selected to be inertial frame, denoted as  $N$ , ball reference frame is represented by  $M$ , ball fixed frame is  $K$  and the body fixed reference frame is  $B$ . Euler angle transformation is used due to its simplicity for representation of body coordinate with respect to the inertial frame.

### A. Dynamical Model

Nonlinear dynamics of the ballbot is briefly described in the following paragraphs. Planar system modeling has been commonly used in the initial studies on the ballbot by dividing the 3D system into the independent planar models by neglecting coupling effects between these models [3], [6]. In this paper, the 3D mathematical model of the ballbot is derived by taking into account all coupling effects under the following assumptions:

- The model consist of five rigid bodies: 1 body with gear heads, 1 ball and 3 omnivheels (see Fig.2).
- The ball moves only horizontally in a pure rolling motion without jumping.
- Contacts between the ball and omnivheels are assumed to be slippage-free.

The system can be described with five DOF (degrees of freedom). 3 DOF for the body rotation  $\Theta_b^N = (\phi_b, \theta_b, \psi_b)$  represent the roll, pitch and yaw angles of the body respectively with respect to  $N$ . 2 DOF for the ball angles

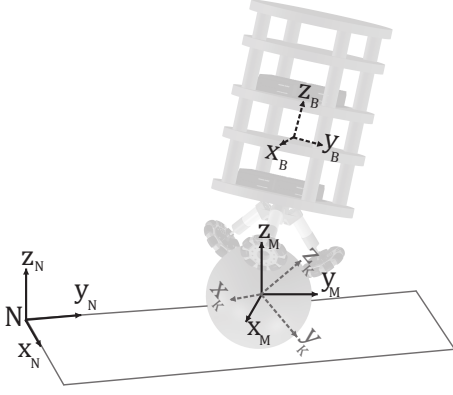


Fig. 2: Coordinate systems (inertial frame  $N$ , ball reference frame  $M$ , ball fixed frame  $K$  and body fixed reference frame  $B$ ).

$\Theta_k^M = (\phi_k, \theta_k)$  with respect to fixed frame  $M$  at the center of the ball. The inputs to the system are the three control torques  $\tau = [\tau_1 \ \tau_2 \ \tau_3]^T$  which are generated by the motors and transformed by the omniwheels to the ball and external disturbances  $\tau_d = [\tau_{d_x} \ \tau_{d_y} \ \tau_{d_z}]^T$  acting on the body. The sensors of the ballbot system are chosen as an IMU on the body and encoders for the motors, and assumed that all states can be measured directly.

The dynamics of the ballbot can be derived using the Euler-Lagrange approach simply by calculating the kinetic and potential energy of each body which has previously been performed by [3], [4], [13]. In order to derive the Euler-Lagrange equations of motion, a minimal coordinates  $q$  is selected to include following set of variables that fully defines the system:

$$q = [\phi_b \ \theta_b \ \psi_b \ \phi_k \ \theta_k]^T \quad (1)$$

This method defines the Lagrangian  $L(q, \dot{q})$  which is simply the sum of the kinetic energy minus the potential energy of the system, as shown in (2).

$$L(q, \dot{q}) = T_b + T_k + T_{w_1} + T_{w_2} + T_{w_3} - V_b \quad (2)$$

Where  $T_b$  and  $T_k$  represent the sum of the translational and rotational kinetic energies of the body and the ball respectively.  $T_{w_i}$  represents rotational kinetic energies of omniwheels and  $V_b$  is the sum of the potential energy of the body and omniwheels.

The equations of motion are derived by solving the Lagrange equation in time  $t$  to find the torques that directly control the minimal coordinates  $\tau_q$  which are related to the motor torques  $\tau$  with kinematic relations.

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} \right)^T - \left( \frac{\partial L}{\partial q} \right) = \tau_q(q, \tau) \quad (3)$$

Overall, resulting in non-linear equations of motion in compact form is obtained as

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = \tau_q \quad (4)$$

All energy equations described above are calculated, as well as solution for the Lagrange equation is found by using

a dynamic symbol manipulation software package Autolev<sup>TM</sup> [16]. The equations of motion for the system described above are obtained from software as a set of nonlinear, differential equations consist of many terms. Therefore, the overall equations of motion are provided in an external source<sup>1</sup>.

### III. CONTROL SCHEMES

In this section, the design of a balancing controller is discussed by using two different approaches acceleration feedback (AFB) control and LMI based state feedback control for the system developed in Section II. Also a brief description of the position controller is provided.

#### A. Balancing Control: Acceleration Feedback Approach

In this work, the inner acceleration control approach detailed in [17] is implemented as part of cascaded position, velocity and current control loops as depicted in Figure 3. The goal of acceleration control is to improve the stabilization performance of the ballbot system by rejecting external disturbances. The position loop produces reference signals for the velocity loop, which in turn creates reference signals for the current loop. The success of stabilization control largely depends on reliable acceleration feedback, noise-adjusted angular accelerations, provided by sensor fusion algorithm, are used as feedback in the inner current control loops.

In control loops following notations are used to describe motion vectors,  $\Theta = [\phi \ \theta \ \psi]^T$  represent Euler angles, Euler rates described by  $\Omega = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$ , and Euler accelerations are denoted by  $\Gamma = [\ddot{\phi} \ \ddot{\theta} \ \ddot{\psi}]^T$ .

Higher acceleration gain,  $K_\gamma$ , adds more electronic inertia to the physical inertia of the total system. Thus, the overall system exhibits high dynamic stiffness and has better disturbance rejection. The increase in effective inertia reduces the speed of the system's response. In order to preserve the loop gain, one should scale up the control loop gains by the factor  $(1 + K_\gamma)$  in (6).

The following PID controllers generate reference velocities for the velocity control loops, for the orientation errors are defined as  $e_\Theta$ :

$$\Omega_{ref} = K_{p_\Theta} e_\Theta + K_{i_\Theta} \int e_\Theta dt + K_{d_\Theta} \dot{e}_\Theta \quad (5)$$

Velocity and current controls are designed as PI controllers and reference currents are generated as follows, where  $e_\Omega$  represents the velocity errors in the Euler rates.

$$I_{ref} = (1 + K_\gamma) K_{p_\Omega} e_\Omega + (1 + K_\gamma) K_{i_\Omega} \int e_\Omega dt \quad (6)$$

Herewith, the balancing torque is calculated as

$$\tau_{k_b} = K_{p_I} e_I + K_{i_I} \int e_I dt \quad (7)$$

where the acceleration feedback comes into picture and current error is defined as:

$$e_I = I_{ref} - \hat{I} \quad (8)$$

$$\hat{I} = K_\gamma \Gamma \quad (9)$$

<sup>1</sup>Detailed equations of motion for ballbot are available at: <http://cvr.sabanciuniv.edu/ballbot>

## B. Balancing Control: An LMI Approach

Secondly, a linear state feedback approach, which is commonly used for ballbot platforms, is extended to the convex optimization problem and detailed below for comparison purpose. In control systems, the performance requirements, can be expressed as bound inequalities. A large class of such problems falls in the category of convex optimization which have been solved using LMI. The main goal of balancing is to maintain the current body angles at the upright position, where the equations of motion are only needed around  $\phi_b \approx \theta_b \approx 0$ . After linearization, let's define a linear feedback controller for the balancing torque where  $K$  is control gain matrix

$$\tau_{k_b} = -Kx \quad (10)$$

and a quadratic Lyapunov function  $V(x) = x^T P x$  to be asymptotic stable for  $P$  is symmetric positive definite. For  $J$  is to be quadratic cost function,

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (11)$$

Where  $Q$  and  $R$  are the control weighting matrices. A controller which minimizes the worst case cost is given by solving the following optimization problem

$$\min tr(P) \quad (12)$$

$$s.t. (A + BK)^T P + P(A + BK) < -Q - K^T R K \quad (13)$$

can be handled as an LMI based convex problem for solving the inequalities by changing variables [15] and employing a Schur complement [18],

$$Y = P^{-1} \quad \text{and} \quad L = KY \quad (14)$$

following convex problem is obtained

$$\max tr(Y) \\ s.t. \left\{ \begin{array}{ccc} -(AY + BL)^T - (AY + BL) & Y & L^T \\ Y & Q^{-1} & 0 \\ L & 0 & R^{-1} \end{array} \right\} \geq 0$$

optimization problem solved using Matlab, which resulted in the following gain matrix  $K$ :

$$K = LY^{-1} \quad (15)$$

## C. Position Control

In the proposed approach, the position controller is constructed according to the absolute position of the ball in the inertial coordinates. Tracking control is mainly focused on the controlling the position and velocity of the ball center. Since the ball moves only horizontally in a pure rolling motion without slippage between the ball and the ground, the horizontal positional dynamics of the ball in the inertial coordinates can be written as

$$v = r_k \begin{bmatrix} s_{\psi_b} & c_{\psi_b} \\ c_{\psi_b} & s_{\psi_b} \end{bmatrix} \Omega_k \quad (16)$$

where  $\Omega_k = [\dot{\phi}_k, \dot{\theta}_k]^T$  is angular velocity of the ball,  $r_k$  is the radius of the ball, and  $c_{\theta_i}$  and  $s_{\theta_i}$  represent cosine and sine

of the  $\theta$  in frame  $i$ , respectively. By taking time derivative of both side and rearranging to find angular acceleration of the ball as  $\Gamma_k = [\ddot{\phi}_k, \ddot{\theta}_k]^T$ ,

$$\Gamma_k = \frac{1}{r_k} \begin{bmatrix} s_{\psi_b} & -c_{\psi_b} \\ c_{\psi_b} & s_{\psi_b} \end{bmatrix} a - \begin{bmatrix} 0 & c_{\psi_b}^2 - s_{\psi_b}^2 \\ 1 & -2c_{\psi_b} s_{\psi_b} \end{bmatrix} \Omega_k \dot{\psi}_b \quad (17)$$

Desired acceleration of the ballbot is constructed from the outputs of position controller as in the typical motion control systems contain cascaded position and velocity loops. The position loop produces reference signals for the velocity loop, which in turn creates desired acceleration  $a_{ref}$  signal for the system defined in the inertial frame. Position and velocity controls are designed as PID and PI respectively.

$$v_{ref} = K_{p_p} e_p + K_{i_p} \int e_p dt + K_{d_p} \dot{e}_p \quad (18)$$

$$a_{ref} = K_{p_v} e_v + K_{i_v} \int e_v dt \quad (19)$$

where  $e_p$  and  $e_v$  represent the position and velocity errors, respectively.  $K_p$ ,  $K_i$  and  $K_d$  are the proportional, integral and derivative control gains for corresponding variables. Then, desired torques on the ball coming from tracking controller can be calculated after doing the transformation detailed above and multiplying by the inertias of the ball,  $I_k$ , as

$$\tau_{k_p} = I_k \Gamma_k \quad (20)$$

The main simulation block diagram of the ballbot and controllers is depicted in Figure 3, where  $x$  is the state of the system,  $x_{ref}$  is the reference state for balancing controller. Additionally,  $\tau_d$  is the unknown disturbance acting directly on body and  $K_\gamma$  is the acceleration feedback gain.

## IV. SIMULATION RESULTS

Simulation results for two different scenarios are presented in this section where the proposed balancing and tracking controls are incorporated. In the first one, self-balancing performance is investigated, while in the second simulation a horizontal cartesian trajectory is tracked by the ballbot. All simulations are implemented in Simulink. Furthermore, external disturbances are generated to illustrate realistic effects of environmental factors in both scenarios. The simulation does not include friction and/or backlash terms that are present in real ballbots. The physical parameters for the simulation purpose are presented in Table I, where these parameters are calculated from a CAD software model by considering appropriate sizing and material selection of our future ballbot design.

### A. Self-Balancing Results

In the first simulation, the body is initially tilted from its vertical position as setting roll angle as  $\phi_b = -20 \text{ deg}$ . During the balancing, the ballbot is exposed to sudden shocks on the body. Disturbance acting on the ballbot is modeled as one point force like human touch. In order to illustrate this case, disturbance torques are modeled as short duration pulses with a magnitude of  $\pm 20 \text{ Nm}$  that are applied on the system between 5 and 10 seconds as shown in Figure 4. Balancing performances of the both PID+AFB control and LMI based state feedback control are shown in Figures 5 and 6.

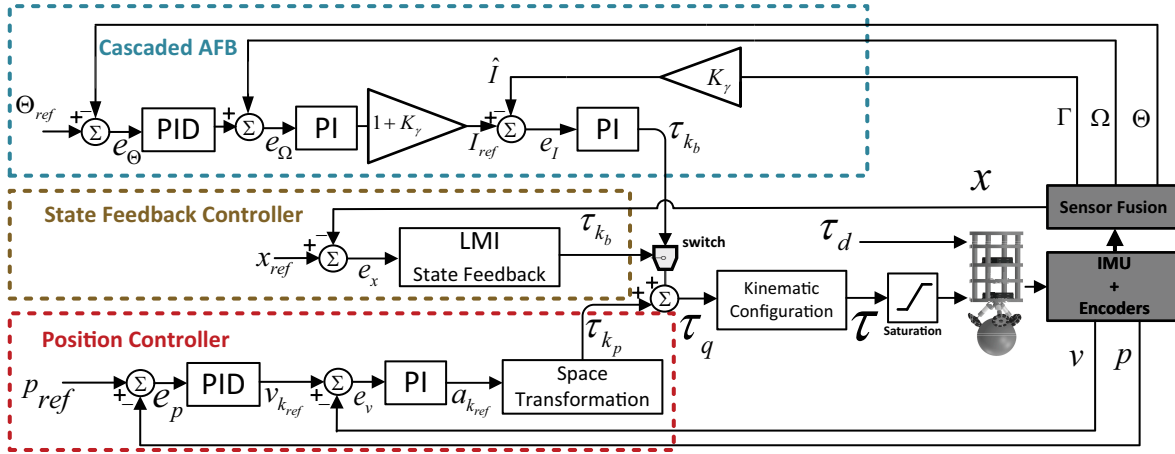


Fig. 3: Block diagram of the balancing controllers augmented with virtual position control.

TABLE I: Definitions of all the simulation parameters.

Parameter	Description	Value
$m_b$	Mass of the body	10 kg
$m_k$	Mass of the ball	3 kg
$m_w$	Mass of each omniwheel	1 kg
$L$	The height of the center of mass of body	0.5 m
$r_k$	Radius of the ball	0.125 m
$r_w$	Radius of each omniwheel	0.05 m
$I_x, I_y$	Inertia of the body around x and y axis	2 kgm <sup>2</sup>
$I_z$	Inertia of the body around z axis	0.1 kgm <sup>2</sup>
$I_k$	Inertia of the ball	0.025 kgm <sup>2</sup>
$I_w$	Inertia of omniwheels	0.0025 kgm <sup>2</sup>
$\alpha$	Omn wheel-ball configuration angle	40°
$\beta$	Angle between omniwheels	120°

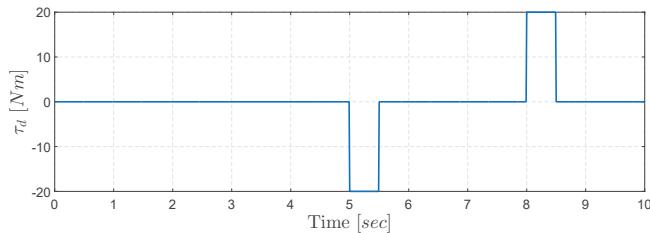


Fig. 4: External disturbances applied on the body through roll axis.

Body roll angle converged to zero from initially tilted position with fast transient response for both controller but LMI based controller resulted was a bit slower. When the acceleration feedback is not used, body orientation can not be preserved due to sudden shocks between 5 – 10 sec. The plots show that with AFB controller the ballbot has successfully stayed upright when subjected to the push during balancing. Body roll angle remained around zero with an maximum error 0.35 deg. However, LMI state feedback controller could not reject disturbances and cause the ballbot to incline through roll and pitch axes, then it takes around 2 seconds to settle back to upright position. The disturbance resulted in a maximum deviation from upright of approximately 5.6 deg for roll and 0.45 deg for pitch angles.

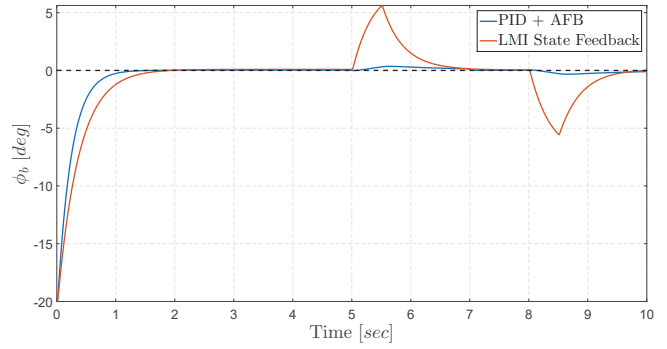


Fig. 5: Simulation results of the self-balancing: Body roll angle  $\phi_b$ .

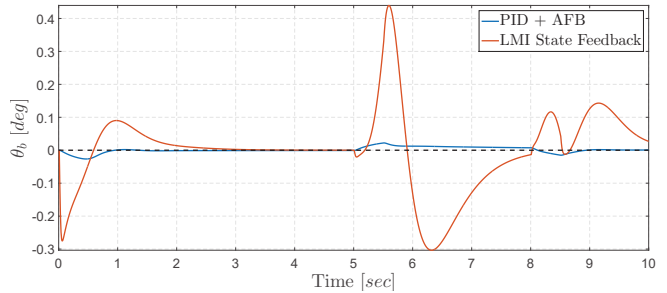


Fig. 6: Simulation results of the self-balancing: Body pitch angle  $\theta_b$ .

### B. Trajectory Tracking Results

In the second simulation, the ballbot tracked a time-varying trajectory starting from the origin of the inertial frame. During trajectory tracking, the ballbot has to perform self-balancing. In order to prevent rapid acceleration that makes the body to attempt aggressively maneuvers toward the reference, desired trajectories are generated by using virtual time by changing real time scale. In doing so, velocity and acceleration references can be slowed down and smoothed trajectories are generated with zero initial velocities and accelerations, where the maximum velocity is 0.1 m/s and maximum acceleration

is  $0.014 \text{ m/s}^2$ . With a well designed reference trajectory, the ballbot can accomplish a given task more stable. In this scenario, initial conditions for all states are selected as zero. During trajectory tracking, the ballbot is exposed to random generated small external disturbances to show effectiveness of the proposed AFB controller.

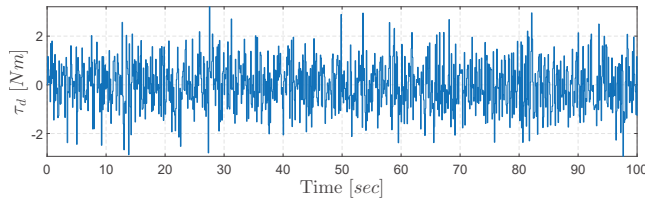


Fig. 7: External disturbances applied on the body through roll axis.

Tracking response on  $XY$  plane is presented in the following figure. The ballbot tracked the reference trajectory in a much smoother way when the AFB is used with the cascaded position controller. In other words, random external disturbances that are given in Figure 7 are rejected. In contrast, LMI based state feedback controller is failed to follow reference properly and resulted in a drift error almost up to 18 cm. Rms of the drift error is found as 5 cm when LMI based state feedback control is used, for AFB control it decreased about 80% to 1 cm.

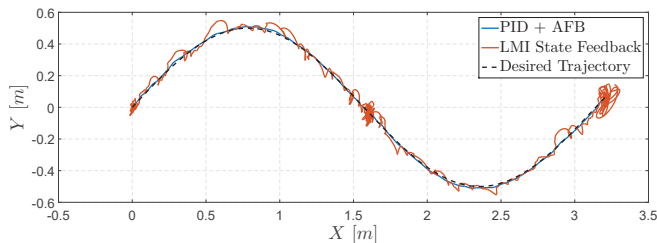


Fig. 8: Simulation results of the tracking: 2D position.

## V. CONCLUSION AND FUTURE WORK

This paper presented self-balancing and position control of a single spherical wheeled mobile platform driven by three omniwheels. The underactuated 3D model of the ballbot platform is derived by considering rigid body assumptions. Acceleration signals are incorporated as feedback to the cascaded position, velocity and current control loops which are used to provide robust performance and compared with a LMI based linear feedback controller. Numerical examples are illustrated via simulations to present the comparison of two control methods in two scenarios subject to different external disturbances. Self-balancing results were quite satisfying. Much better stabilization performance is achieved by acceleration feedback (AFB) controller which successfully stabilized ballbot from a tilted initial condition and maintained upright orientation despite disturbances. In the second scenario, position controller followed the reference trajectory in a cooperation with self-balancing controller under relatively small disturbance. However tracking with AFB balancing control was much more smooth as compared to state feedback. As a future work, real-time motion experiments can be conducted to demonstrate the applicability of the developed controllers.

## REFERENCES

- [1] T. Endo and Y. Nakamura, "An omnidirectional vehicle on a basketball," in *Advanced Robotics, 2005. ICAR'05. Proceedings., 12th International Conference on*. IEEE, 2005, pp. 573–578.
- [2] L. Havasi, "Errosphere: an equilibror robot," in *Control and Automation, 2005. ICCA'05. International Conference on*, vol. 2. IEEE, 2005, pp. 971–976.
- [3] T. Lauwers, G. A. Kantor, and R. L. Hollis, "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.
- [4] U. Nagarajan, A. Mampetta, G. A. Kantor, and R. L. Hollis, "State transition, balancing, station keeping, and yaw control for a dynamically stable single spherical wheel mobile robot," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 998–1003.
- [5] U. Nagarajan, B. Kim, and R. Hollis, "Planning in high-dimensional shape space for a single-wheeled balancing mobile robot with arms," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 130–135.
- [6] 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.
- [7] L. Hertig, D. Schindler, M. Bloesch, C. D. Remy, and R. Siegwart, "Unified state estimation for a ballbot," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2471–2476.
- [8] 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. Available at: <http://www.rezero.ethz.ch>
- [9] M. Kumagai, "Development of a ball drive unit using partially sliding rollersan alternative mechanism for semi-omnidirectional motion," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 3352–3357.
- [10] A. Bhatia, M. Kumagai, and R. Hollis, "Six-stator spherical induction motor for balancing mobile robots," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 226–231.
- [11] mObi, Bossa Nova Robotics, December 2012. Available at: <http://www.bnrobotics.com/mobi/>.
- [12] U. Nagarajan, G. Kantor, and R. L. Hollis, "Trajectory planning and control of an underactuated dynamically stable single spherical wheeled mobile robot," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 3743–3748.
- [13] A. N. Inal, Ö. Morgül, and U. Saranlı, "A 3d dynamic model of a spherical wheeled self-balancing robot," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 5381–5386.
- [14] P. B. Schmidt and R. D. Lorenz, "Design principles and implementation of acceleration feedback to improve performance of dc drives," *Industry Applications, IEEE Transactions on*, vol. 28, no. 3, pp. 594–599, 1992.,
- [15] S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994, vol. 15.
- [16] T. R. Kane and D. A. Levinson, *Solution Manual for Problem Sets for Dynamics Online: Theory and Implementation with AUTOLEV*. OnLine Dynamics, 2000.
- [17] Y. Hori, "Disturbance suppression on an acceleration control type dc servo system," in *Power Electronics Specialists Conference, 1988. PESC'88 Record., 19th Annual IEEE*. IEEE, 1988, pp. 222–229.
- [18] A. Laub, "A schur method for solving algebraic riccati equations," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 913–921, Dec 1979.