Balancing and Transferring Control of a Ball Segway Using a Double-Loop Approach

DINH BA PHAM, HYUNG KIM, JAEJUN KIM, and SOON-GEUL LEE

Mobile robots typically have more than three contacts with the ground, including at least two independent driving wheels, to be statically stable with a wide base for a large support polygon. These robots can rotate at any point but cannot make an instantaneous movement in every direction. They cannot move freely even in an open space when they are crowded with people in a common human-coexisting environment. Moreover, these robots have other limitations, such as minimum passage width and minimum turning radius.

To overcome these types of motion limitations, numerous studies have been conducted on developing omnidirectional mobile robots. One development is a ballbot, which is intended to be dynamically stable and designed to balance on a single spherical wheel (ball) while moving in every direction. (See "Summary.") Generally, a ballbot has a body that balances on top of a ball and forms an under-actuated system; thus, the robot has more degrees of freedom (DOF) than independent control inputs. Actuators directly drive the ball, and the body has no direct control. The body is maintained upright at approximately its unstable equilibrium point by controlling the ball, which is similar to controlling an inverted pendulum. These characteristics of the ballbot result in several challenges in the balancing and transferring control. See "Balancing and Transferring Control Problems for a Ball Segway" for more details on balancing and transferring control problems.

The first ballbot, developed in 2006, was a mobile robot balancing on a ball [1]. This ballbot was designed with human-like dimensions that enabled interaction with a person at eye level, so the ballbot could effectively interact with humans and transfer and rotate in a populated environment [2]. This study used an inverse mouse-ball drive mechanism for the ball and presented a linear control law to ensure stabilization and basic motion control.

Ball IP [3], which was introduced in 2009, is another version of a ballbot that balances on a spherical wheel. The robot uses a drive mechanism that comprises three omnidirectional wheels with stepping motors, which are in contact with a spherical wheel that enables control affordance and friction reduction. A typical proportional-derivative (PD) feedback controller is used to control balancing and transferring actions. The robot can transport loads in any direction, but it can only support small external loads. Rezero [4], a type of ballbot, was developed in 2010 and used three omnidirectional wheels to drive a spherical wheel, thereby resulting in smooth mobility on the floor. Several other ballbots [5], [6] equipped with a three-omnidirectional-wheel drive mechanism were also developed but were also limited in motion.

BB rider [7] is a ballbot consisting of four omnidirectional wheels for driving a spherical tire. The robot has a simple wheelchair structure that enables it to transport a person. The kinematic model and force relationship of the omnidirectional wheels and the spherical tire required and generated by the motors were also formulated. Although the BB rider is a personal carrier ball robot, experimental results were not included in this study.

Summary

A ball segway is developed with the question of whether a people-carrier robot could turn 360° or drive itself freely with a spherical wheel. The ball segway is similar to a ballbot, and it forms a spherical inverted pendulum with a car-like structure mounted on top that moves on a ball. The vehicle described in this article is the world's first ballbot-type people-carrying robot that is sufficiently large enough for a human to ride. By using the spherical wheel, it is very easy to implement parallel parking and also advantageous to reduce the uneven wear of the wheel. Performance in cornering and turning is also significantly better than that of conventional wheels. An Intel® NUC PC is used as the main controller, and Windows Embedded 7 is used as the operating system. The control algorithm is realized with multithread programming in Visual Studio C++ to ensure the control interval of 15 ms. Numerical and experimental results show that the robot can maintain its balance, track the desired trajectory, and also simultaneously rotate around the vertical axis.

Digital Object Identifier 10.1109/MCS.2017.2786444
Date of publication: 16 March 2018

1066-033X/18©2018IEEE

Balancing and Transferring Control Problems for a Ball Segway

The ball segway is a spatial, omnidirectional-wheeled, inverted pendulum robot with a rigid body atop a single ball. One of the interesting but challenging features of the ball segway is its unstable, underactuated, and second-order, nonholonomic velocity constraints. These constraints restrict the family of trajectories that configurations can follow. The underactuated and unstable dynamics of the ball segway result in challenging planning and control tasks. Moreover, strong coupling is observed between the dynamics of the ball and the body. If the ball is moved, then the body falls. If the body is held upright by moving the ball, then the ball is not in its planned position on the floor. Therefore, dynamic motion for the ball segway is a challenging task in obtaining a goal position while balancing on the floor.

The principle used to balance the ball segway is considerably similar to the principle used for balancing a stick. When the ball segway tilts, a balancing controller provides an action to move the ball in the direction of the tilt; thus, the robot stays upright. The robot also needs to accelerate by tilting in the moving direction to transfer in any direction on the floor. However, the robot cannot directly control its tilt body angles. Thus, the balance and movement of the ball segway are achieved by only maneuvering the ball on the floor. For example, in the case of carrying a person, the ball segway should maintain balance by tilting back and tilting into the ramp while ascending or descending to a steep slope.

Compared with previous ballbots [2]–[3], [12], [15], the ball segway has a car-like structure (as shown in Figure S1) and is the system with a heavy weight; the mass and moments of inertia of the ball segway are changeable when ridden. A large amount of mass of the entire ball segway system causes more difficulty in the firm control of the ball segway movement from the initial position to the desired position while balancing. Driving motors can generate enough torque to overcome the static friction of the system and compensate large inertial force during stopping. However, uncertainties in the ball segway parameters, such as the moment of inertia, the mass center of the body, the friction factors between the body and the ball and between the ball and the floor, floor conditions, and external disturbances, affect the balance and movement of the ball segway.

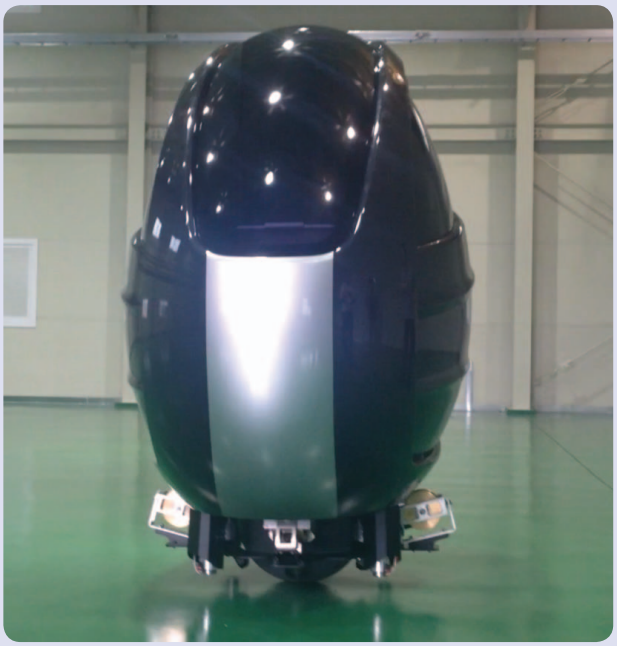


FIGURE S1 A ball segway balancing on the floor. It is a ballbot-type carrier robot that can accommodate a person to ride on a seat of a car-like structure mounted on a single ball.

The design of a ball segway aims to transport humans, who need to physically interact with the robot in the actual environment. Thus, the robot requires stringent balancing under any condition. The ball segway is naturally reactive because physical interactions are detected as disturbances to its balancing behavior. The ball segway is also equipped with a control moment gyroscope cluster to ensure balance when a person rides on it. All these characteristics of the ball segway show that balancing and transferring controls are a challenging task when developing the ball segway.

Together with the development of ballbot structures and types (as shown in Figure 1), control problems (which vary from classical control methods [6], [8], [9] to modern techniques with single-loop [3], [10] and multiloop [1], [2], [11], [12] control systems) were discussed by many researchers. A linear-quadratic regulator (LQR) control law was found by using a linear state-space model to control balancing and station keeping [10]. A robust control based on the sliding-mode technique [13] was presented to control balancing and transferring a ballbot with an inverse mouse-ball drive mechanism. Furthermore, a simulation research was reported. Another robust control [14] based on linear matrix inequalities was used as an approach to manage parametric uncertainties, thereby providing robustness for balancing a ballbot. Moreover, intelligent controls, such as fuzzy logic control [15]–[18], were utilized to control balancing and transferring a ballbot. In [19], the author presented an intelligent consensus-based cooperative formation control using a recurrent fuzzy wavelet cerebellar model articulation controller for a team of uncertain multiple ballbots. However, experiments were not included in this study.

With regard to multiloop control schemes, a double-loop control system, including an LQR outer-loop controller and a proportional-integral (PI) inner-loop controller, was presented to control the self-balancing and transferring functions of a ballbot [11]. An LQR with an integral controller

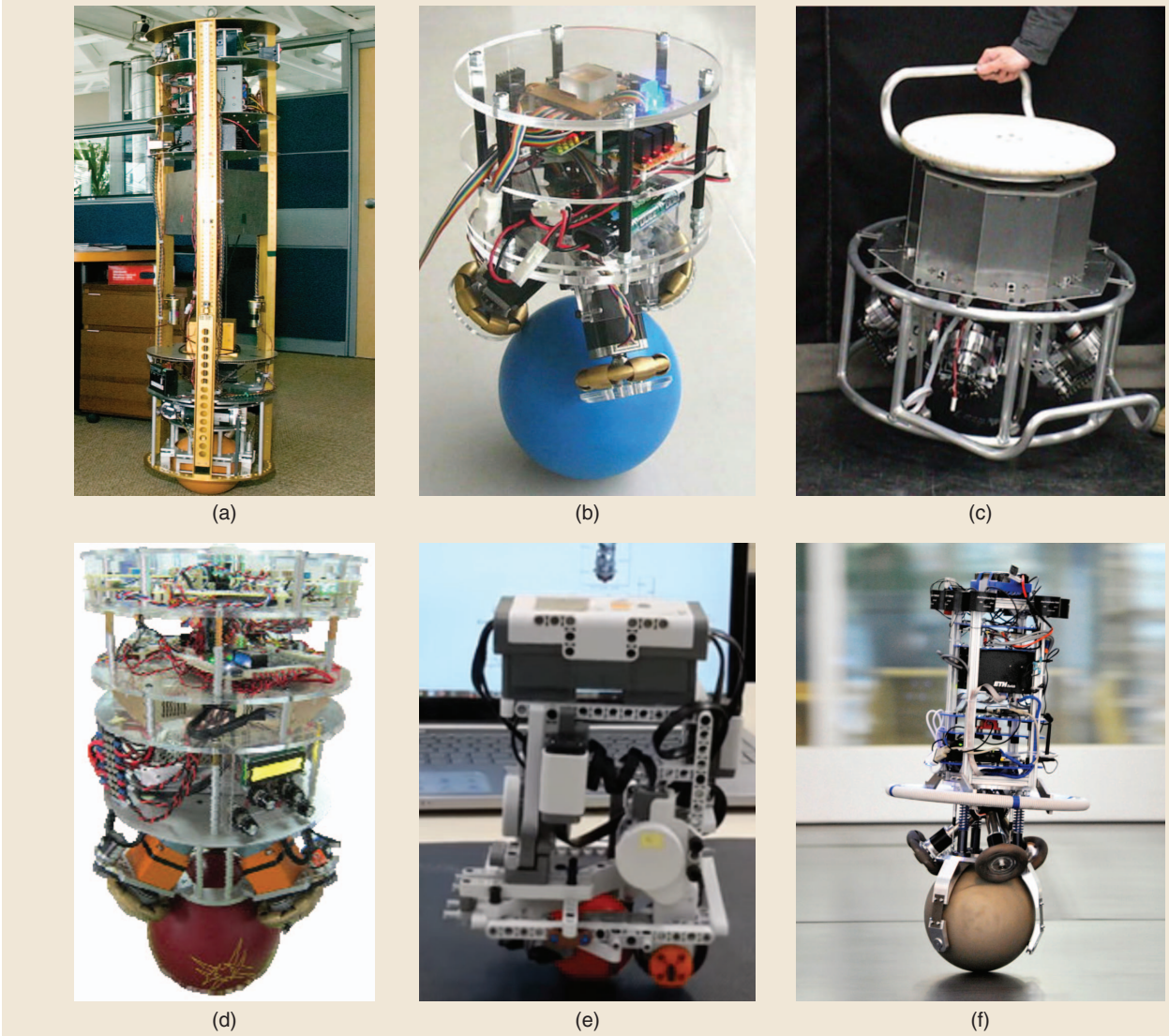


FIGURE 1 Various types of ballbots. (a) The first ballbot with an inverse mouse-ball drive mechanism was developed in 2006 at Carnegie Mellon University, United States. (Used with permission from [1], Ralph Hollis, Carnegie Mellon University.) (b) A ballbot with a three-omnidirectional-wheel drive mechanism was developed in 2009 at Tohoku Gakuin University, Japan. (Used with permission from [3], Massaki Kumagai, Tohoku Gakuin University.) (c) The simple wheelchair structure of the BB rider enables it to transport humans. (From [7].) (d) The National Chung Hsing University, Taiwan, developed a ballbot in 2012 with three omnidirectional wheels to drive a ball. (Used with permission from [10], Ching-Chih, National Chung Hsing University.) (e) The ballbot using LEGO NXT is a small robot of approximately 20 cm high, with only two wheels to drive a ball. It was developed at the University of Adelaide, Australia, in 2009. (Adapted from "Design and Build a ballbot," by J. Fong, S. Uppill, 2009 Mechatronics honors project, University of Adelaide, used with permission from Ben Cazzolato, The University of Adelaide.) (f) Rezero, a type of ballbot, was developed at Autonomous Systems Lab, ETH Zurich, Switzerland in 2010. The robot is equipped with three omnidirectional wheels to drive a ball. (Adapted from [4].)

was applied to balance a new double-level, ball-riding robot, which consists of upper and lower ball-riding subsystems [12]. The upper ball-riding subsystem rode on top of the lower one. Simulations and experiments showed the balancing performance of the proposed controller.

As discussed in "Summary," the significant contributions of this study are twofold; one is deriving a completely dynamic model of a ball segway by incorporating the kinematics of an inverse Atlas spherical motion platform, and the other is designing and testing the balance, robustness, and transfer controllers based on an improved double-loop control strategy. The improved double-loop strategy includes a PI-plus-feedforward inner-loop control and an LQR outer-loop control that enable the ball segway to balance and transfer to the desired position. A feedforward compensation is added for the stability of the ball segway even with heavy weight and bulk, unknown friction, parameter variations, and external disturbances.

The results presented in this study are intended to help bridge the gap between the theory of linear controls and its application to underactuated mechanical systems, such as ball segways, with emphasis on using a double-loop control system with an optimal LQR control law. In the double-loop control system, additional control is considered the feedforward control to eliminate almost all uncertainties caused by undefined frictions, modeling, and system parameters. Thus, with a double-loop control structure, an inner-loop controller provides direct control input in balancing and transferring the ball segway. An outer-loop controller is exploited to withstand the effects of external disturbances. Moreover, the ball segway is a representative underactuated system used for the experimental implementation.

The ball segway is programmed in Matlab and Visual Studio C++. The completely nonlinear dynamic model of the two-dimensional (2-D) ball segway is built using a Symbolic Math Toolbox in Matlab. The nonlinear model is linearized around an equilibrium point, and this linear model is used to synthesize the LQR and PI double-loop control algorithm. The responses of the ball segway system are investigated through the simulation in Matlab and Simulink. Numerical simulations verify the validity and stability of the double-loop control system. In practice, the robot controller is programmed in Visual Studio C++ to utilize the advantages of the programming language, including fast data transfer and easy connection to hardware devices, such as sensors, wireless joysticks, and driving motors. Finally, experimental results are presented to demonstrate the balancing and transferring behaviors of the controllers in the presence of external disturbances and undefined friction.

SYSTEM CONFIGURATION

Ball Segway Platform Design

The ball segway shown in Figure 2(a) [and Figure 18(d)] is an advertising ballbot platform for a tire company [20]. This study aims to design a ball segway that transports a person. The ball segway is composed of a drive mechanism and a body part with a cabin in which a person can ride. Figure 2(a) also shows a body frame made of aluminum and a plastic cover.

The ball segway is designed to be entirely self-contained. Power is supplied by batteries, and computation is performed by a minicomputer. The driving motors with encoders that feed the motor shaft position are directly connected to the computer. The ball segway is equipped with an inertial measurement unit (IMU) for measuring angles and angular rates of the body.

Drive Mechanism Design

The drive mechanism of the ball segway, as presented in Figure 2(b), uses three single-row omnidirectional wheels with an inverse Atlas spherical motion platform. Each single-row omnidirectional wheel is driven by a motor through a reduction gear [21]. The motors are symmetrically mounted at intervals of 120°, as shown in Figure 2(b). The shaft of each single-row omnidirectional wheel is perpendicular to the tangent plane of the spherical wheel. The three omnidirectional wheels provide the motion that allows the ball segway to rotate, balance, and transfer accurately in any direction in an actual environment without constraints.

A frame is used to mount the drive mechanism properly, guarantee and synchronize the contact between the omnidirectional wheels and the spherical wheel, and ensure that the omnidirectional wheels transmit the same amount of force and velocity to the spherical wheel.

The spherical wheel is made of a hollow aluminum sphere that can bear the weight of the robot body and the person riding in it. The thick rubber clad over the spherical aluminum wheel provides sufficient friction between the omnidirectional wheels and the spherical wheel and between the spherical wheel and the floor for transferring and pivoting around the vertical axis. Table 1 lists the specifications of the developed rideable ball segway.

SYSTEM MODEL

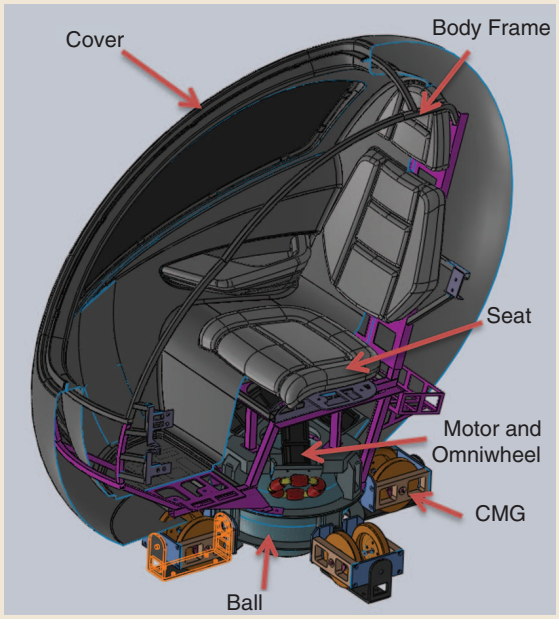
A mathematical model is required to design a proper controller that achieves the balance and desired transfer control performance. The ball segway system has five DOFs, of which two are dedicated to the translation of the ball and three to the rotation of the body. Therefore, the minimal coordinate vector is defined as $\mathbf{q} = [x_k \;\; y_k \;\; \theta_x \;\; \theta_y \;\; \theta_z]^T$, where x_k and y_k indicate the position of the ball along the x- and y-axes, respectively, and θ_x, θ_y, and θ_z are body angles (in Tait–Bryan angles) along the y–z, x–z, and x–y planes, respectively.

As illustrated in Figure 3, the following six coordinate frames are defined for system modeling. The global inertial reference frame is denoted as I. Reference frame 1 is obtained by translating the global reference frame I by x_k along the x-axis. Reference frame 2 is achieved by translating the reference frame 1 by y_k along the y-axis. Reference frame 3 is obtained by turning around the z-axis of reference frame 2 by θ_z. Reference frame 4 is derived with the rotation of reference frame 3 around its y-axis by θ_y. Finally, the body-fixed reference frame 5 is derived with the rotation of reference frame 4 around its x-axis by θ_x.

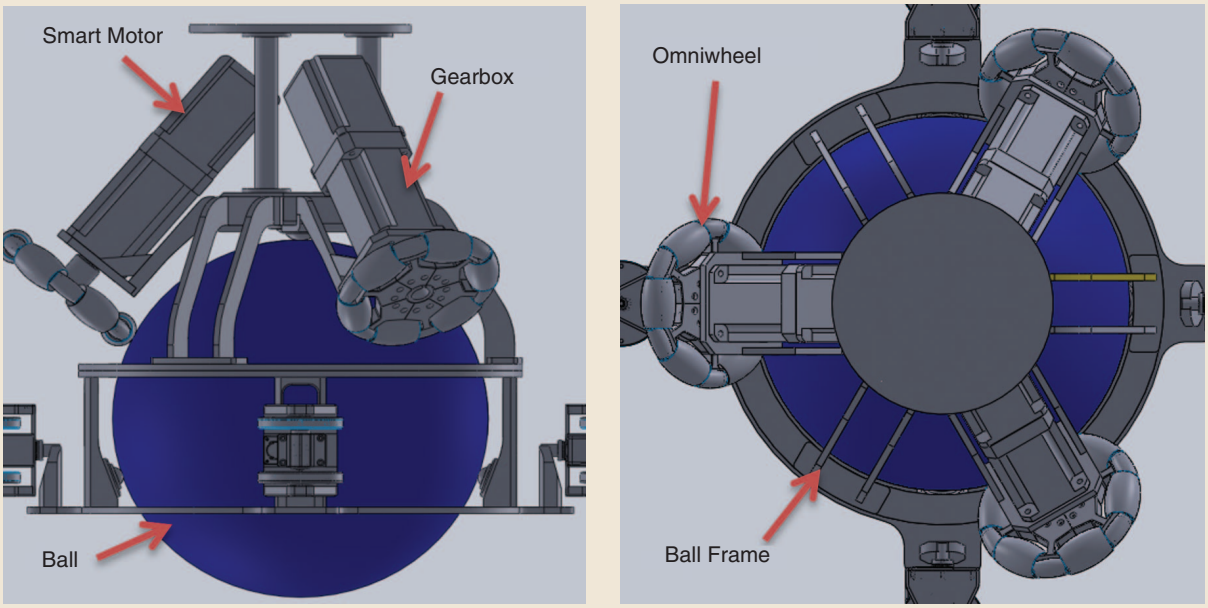
Kinematic Model

The general kinematics of the ball segway is a relationship among $\dot{\boldsymbol{\varphi}}_x = [\dot{\varphi}_x \;\; \dot{\varphi}_y \;\; \dot{\varphi}_z]^T$, the angular velocity of the spherical wheel, and $\dot{\psi}_i \, (i = 1,2,3)$, the angular velocities of the three omnidirectional wheels relative to the body frame. The kinematic relationship of dual-row omnidirectional wheels is discussed in [22]. Meanwhile, single-row omnidirectional wheels are used for the ball segway. The shafts of the omnidirectional wheels are separated by 120° in the x–y plane, and each shaft slopes downward by α.

The main assumptions for the simplification of the system are:



(a)



(b)

FIGURE 2 The structure of the ball segway, which is designed to transport a person. (a) An overview of the ball segway. (b) The drive mechanism of the ball segway, including three omnidirectional wheels acting on a ball, and four control moment gyroscopes (CMGs) to control balancing, transferring, and rotating.

- The body and floor have no deformation.
- No slip occurs between the ball and the floor and between the ball and the omnidirectional wheels.
- The contacts between the ball and the ground and between the omnidirectional wheels and the ball are point contacts.

To meet the no-slip condition between the ball and the omnidirectional wheels, the projection of the velocities of the ball at all contact points in the actuation direction of the omnidirectional wheel should be equal, that is,

$$\dot{\psi}_i r_w = (\dot{\boldsymbol{\varphi}}_k \times \mathbf{P}_i)\mathbf{u}_{wi}, \quad i = 1,2,3, \tag{1}$$

where $\mathbf{P}_i$ is the position vector from the center of the ball to the contact point i of the ith omnidirectional wheel to the

Table 1 Specifications of the ball segway. To estimate the ball segway system parameters, the system can be separated into five rigid bodies: the ball, three omnidirectional wheels, and main body. The ball is modeled as a homogeneous sphere. The omnidirectional wheels attached to the body are modeled as disks rotating about their shafts. The main body is approximated by a solid cuboid with upper and lower parts, with different densities to approximate the nonuniform mass distribution. A physical model of the ball segway system is also built in the SolidWorks software so that the system parameters can be determined in a different way. Therefore, applying and comparing the results of the two aforementioned methods can reduce errors and ensure the accuracy of the system parameters.

Parameter	Symbol	Value	Units
Mass of the body	m_a	68	kg
Body center of the mass height of the body	l	0.38	m
Moment of inertia of the body about the x-axis	I_x	12.1	kg-m^2
Moment of inertia of the body about the y-axis	I_y	11.67	kg-m^2
Moment of inertia of the body about the z-axis	I_z	1.08	kg-m^2
Radius of the omnidirectional wheel	r_w	0.1	m
Moment of inertia of the omnidirectional wheel	I_w	0.26	kg-m^2
Mass of the ball	m_k	19.6	kg
Radius of the ball	r_k	0.22	m
Moment of inertia of the ball	I_k	0.38	kg-m^2
Zenith angle	α	65.5	°

ball in the body frame, as indicated in Figure 4; $\mathbf{u}_{wi}$ is the vector of the direction cosines of the ith omnidirectional wheel contact point velocities in the actuation directions; and r_w is the radius of the omnidirectional wheel.

Therefore, the angular velocities of the omnidirectional wheels are expressed as

$$\dot{\psi}_i = \frac{1}{r_w}(\dot{\boldsymbol{\varphi}}_k \times \mathbf{P}_i)\mathbf{u}_{wi}, \quad i = 1,2,3. \tag{2}$$

As shown in Figure 4, the position vector of each contact point $\mathbf{P}_i$ is

$$\begin{aligned} \mathbf{P}_1 &= [r_k \sin\alpha \;\; 0 \;\; r_k \cos\alpha]^T, \\ \mathbf{P}_{2,3} &= \left[-\frac{1}{2} r_k \sin\alpha \;\; \pm\frac{\sqrt{3}}{2} r_k \sin\alpha \;\; r_k \cos\alpha\right]^T, \end{aligned} \tag{3}$$

where r_k is the radius of the ball.

Furthermore, the direction cosine vectors that provide the corresponding direction of the speed at the contact point between the omnidirectional wheels and the ball are

$$\begin{aligned} \mathbf{u}_{w1} &= [0 \;\; 1 \;\; 0]^T, \\ \mathbf{u}_{w2,3} &= \left[\mp\frac{\sqrt{3}}{2} \;\; -\frac{1}{2} \;\; 0\right]^T. \end{aligned} \tag{4}$$

Substituting (3) and (4) into (2) yields

$$[\dot{\varphi}_x \;\; \dot{\varphi}_y \;\; \dot{\varphi}_z]^T = \mathbf{J}(\alpha)[\dot{\psi}_1 \;\; \dot{\psi}_2 \;\; \dot{\psi}_3]^T, \tag{5}$$

where $\mathbf{J}(\alpha)$ is the Jacobian matrix, that is,

$$\mathbf{J}(\alpha) = \begin{bmatrix} -\frac{2r_w}{3r_k\cos\alpha} & \frac{r_w}{3r_k\cos\alpha} & \frac{r_w}{3r_k\cos\alpha} \\ 0 & -\frac{\sqrt{3}\,r_w}{3r_k\cos\alpha} & \frac{\sqrt{3}\,r_w}{3r_k\cos\alpha} \\ \frac{r_w}{3r_k\sin\alpha} & \frac{r_w}{3r_k\sin\alpha} & \frac{r_w}{3r_k\sin\alpha} \end{bmatrix}. \tag{6}$$

The Jacobian matrix (6) represents the effect of the zenith angle α on the motion relationship between the ball and the three single-row omnidirectional wheels.

Dynamic Model

The three-dimensional (3-D) motion of the ball segway can be decoupled into three 2-D motions, that is, one in the x–z plane, one in the y–z plane, and another in the x–y plane. Each 2-D model describes the 2-D motion in 2-D space.

The dynamic equations of the 2-D ball segway in the y–z plane can be obtained from the Euler–Lagrange equation and transformed into matrix form

$$\mathbf{M}(\mathbf{q}_x)\ddot{\mathbf{q}}_x + \mathbf{C}(\mathbf{q}_x, \dot{\mathbf{q}}_x)\dot{\mathbf{q}}_x + \mathbf{D}(\dot{\mathbf{q}}_x) + \mathbf{G}(\mathbf{q}_x) = \mathbf{Q}_x, \tag{7}$$

where the state vector $\mathbf{q}_x$, mass matrix $\mathbf{M}(\mathbf{q}_x)$, gravity vector $\mathbf{G}(\mathbf{q}_x)$, friction vector $\mathbf{D}(\dot{\mathbf{q}}_x)$, vector of control torque along the y-axis $\mathbf{Q}_x$, and Coriolis-centrifugal vector $\mathbf{C}(\mathbf{q}_x, \dot{\mathbf{q}}_x)$ are respectively expressed as

$$\begin{aligned} \mathbf{q}_x &= [y_k \;\; \theta_x]^T, \\ \mathbf{M}(\mathbf{q}_x) &= \begin{bmatrix} m_k + \frac{I_k}{r_k^2} + m_a + \frac{3I_w\cos^2\alpha}{2r_w^2} & \frac{3I_w\cos^2\alpha}{2r_w^2} r_k - m_a l\cos\theta_x \\ \frac{3I_w\cos^2\alpha}{2r_w^2} r_k - m_a l\cos\theta_x & m_a l^2 + \frac{3I_w r_k^2\cos^2\alpha}{2r_w^2} + I_x \end{bmatrix}, \\ \mathbf{G}(\mathbf{q}_x) &= [0 \;\; -m_a g l\sin\theta_x]^T, \\ \mathbf{D}(\dot{\mathbf{q}}_x) &= [b_y\dot{y}_k + d_y sign(\dot{y}_k) \;\; b_{rx}\dot{\theta}_x + d_{rx} sign(\dot{\theta}_x)]^T, \\ \mathbf{Q}_x &= \left[\frac{1}{r_w}\tau_x \;\; \frac{r_k}{r_w}\tau_x\right]^T, \\ \mathbf{C}(\mathbf{q}_x, \dot{\mathbf{q}}_x) &= \begin{bmatrix} 0 & m_a l\dot{\theta}_x\sin\theta_x \\ 0 & 0 \end{bmatrix}, \end{aligned}$$

where τ_x is the equivalent torque of the three driving motors acting on the ball around the x-axis; g is the gravitational acceleration; b_y and b_{rx} are the viscous damping coefficients that model the spherical wheel–ground friction and spherical wheel–body friction in the y–z plane, respectively; and d_y and d_{rx} are the Coulomb friction coefficients

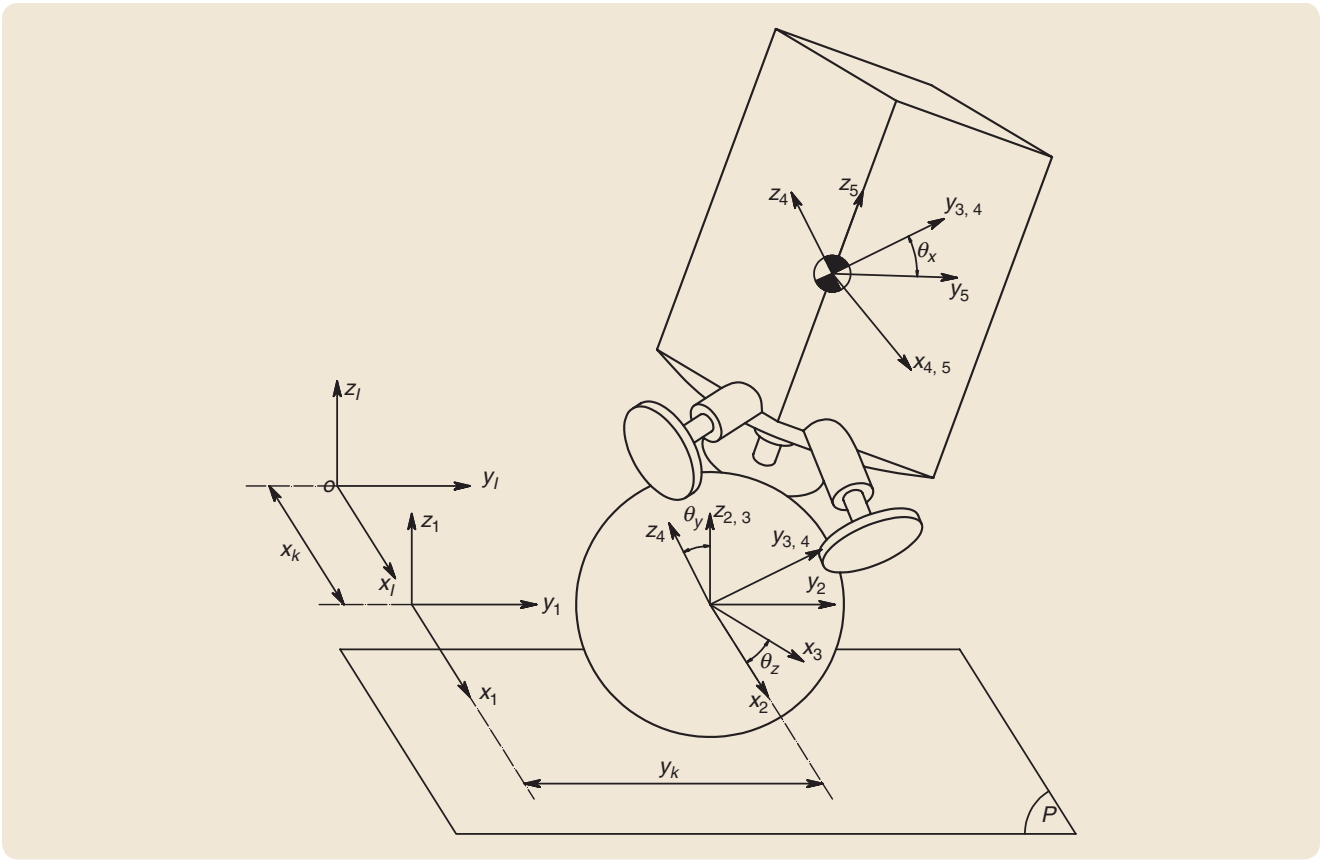


FIGURE 3 A sketch of the three-dimensional (3-D) model and coordinate system. Six coordinate frames are used to derive the motion of the 3-D ball segway. x_k and y_k describe the translational motion of the ball on the floor, and θ_k, θ_y, and θ_z describe the rotational motion of the body in Tait-Bryan angles (a variation on Euler angles).

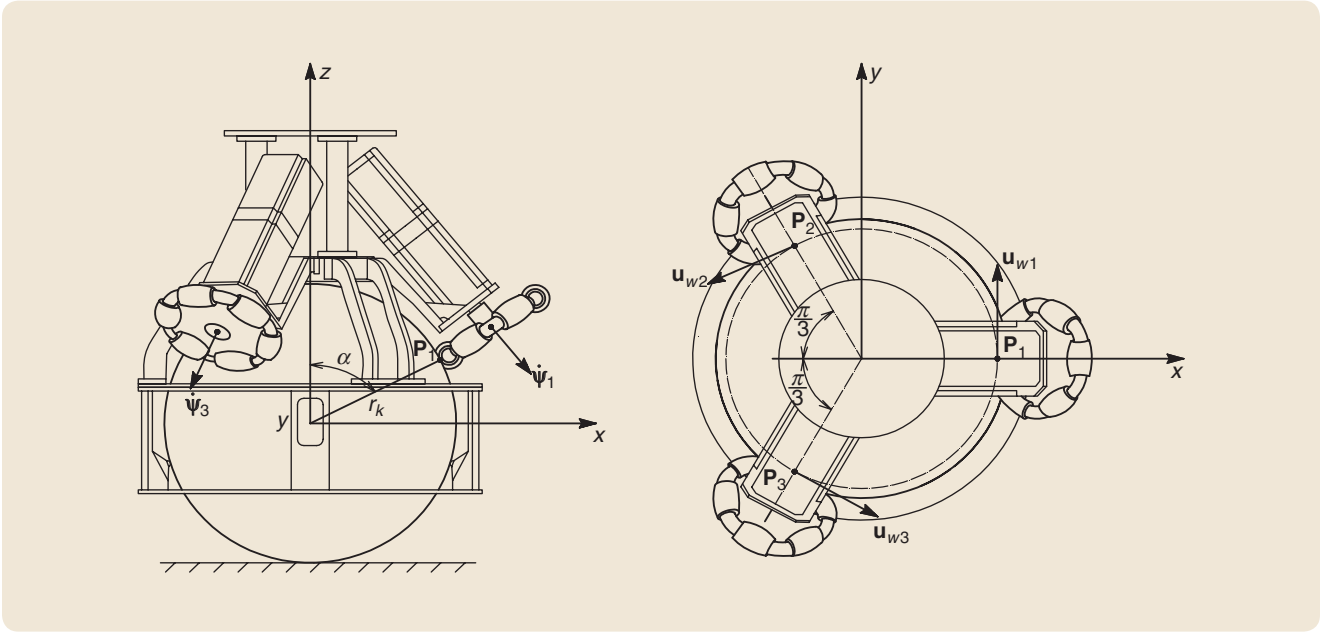


FIGURE 4 A sketch of decomposed angular velocities of three omnidirectional wheels and a ball with radius r_k. Each omnidirectional wheel is fixed to a shaft, thereby making it perpendicular to the tangent plane of the ball. The omnidirectional shafts form the zenith angle α to the vertical axis. The omnidirectional wheels act as friction wheels to transmit the motion to the spherical tire without a resisting motion in the direction orthogonal to the actuation direction. The vectors from the center of the ball to the contact point between the omnidirectional wheels and ball are $\mathbf{P}_1, \mathbf{P}_2$, and $\mathbf{P}_3$ in the body frame. A unit vector $\mathbf{u}_{wi}\,(i = 1, 2, 3)$ denotes tangential direction at each contact point. Based on the no-slip assumption, the vector sum $\sum_i |\dot{\Psi}|\mathbf{u}_{wi}$ decides the rotating speed and direction of the ball moving on the floor.

Nonlinear Dynamic Equation of the Two-Dimensional Ball Segway

The planar model of the ball segway system is composed of the ball, the body, and omnidirectional wheels. The body is a solid cuboid, which models the moment of inertia and center of the body mass. The omnidirectional wheels attached to the body are disks that rotate around their shafts.

The Euler–Lagrange equation and decoupling method are used to derive the dynamic equations of the 2-D ball segway. The physical model of the 2-D ball segway is shown in Figure S2. The selected generalized coordinates of the ball segway system consists of $x_k(t)$, $y_k(t)$, $\theta_x(t)$, $\theta_y(t)$, and $\theta_z(t)$, namely, the ball displacement along the x- and y-axes and the body angles around the x-, y-, and z-axes, respectively. The control inputs, τ_x, τ_y, and τ_z, are the equivalent torques of the three driving motors that act on the ball around the x-, y-, and z-axes. The vector of minimal coordinates in the y–z plane is defined as $\mathbf{q}_x = [y_k \;\; \theta_x]^T$. The dynamic equations are obtained through the Euler–Lagrange equation

$$\frac{d}{dt}\left(\frac{\partial L_x}{\partial \dot{\mathbf{q}}_x}\right) - \frac{\partial L_x}{\partial \mathbf{q}_x} = \mathbf{Q}_x - \mathbf{D}(\dot{\mathbf{q}}_x).$$

The kinetic and potential energies of the ball, body, and omnidirectional wheels should be obtained to describe the dynamics of the ball segway.

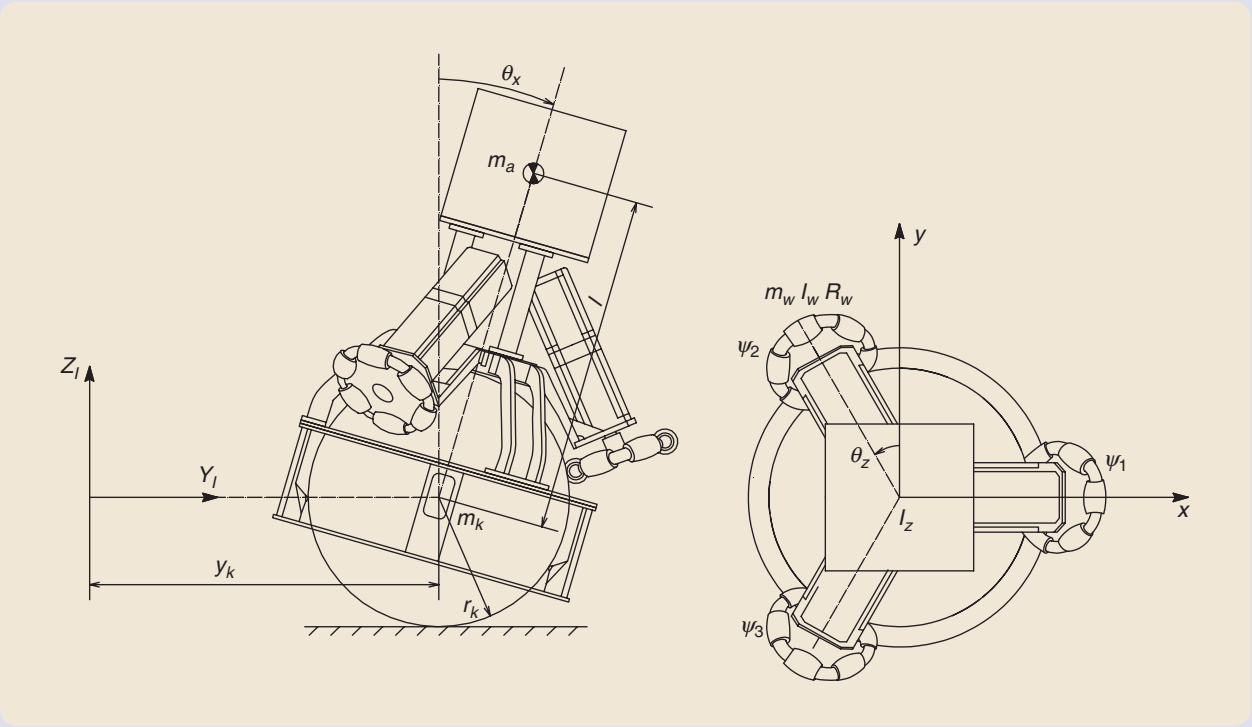


FIGURE S2 A physical system of the 2-D ball segway. One model is for the y–z and x–y planes, and the other is for the x–z plane. The body of the ball segway is modeled as an inverted pendulum with the mass m_a and moments of inertia of the body I_x, I_y, and I_z. The ball is approximated by a homogeneous sphere with the radius r_k and mass m_k. l is the distance between the center of mass of the body and the ball. The vectors of minimal coordinates in the y–z and x–z planes are defined by $\mathbf{q}_x = [y_k \;\; \theta_x]^T$ and $\mathbf{q}_y = [x_k \;\; \theta_y]^T$, where x_k and y_k are the positions of the ball and θ_x, θ_y, and θ_z are the angles of the body.

that model the spherical wheel–ground friction and spherical wheel–body friction in the y–z plane, respectively. The other symbols are presented in Table 1, and "Nonlinear Dynamic Equation of the Two-Dimensional Ball Segway" completely represents the low-level dynamic components.

The dynamic equations derived by the Euler–Lagrange equation in the x–z plane are represented in matrix form

$$\mathbf{M}(\mathbf{q}_y)\ddot{\mathbf{q}}_y + \mathbf{C}(\mathbf{q}_y, \dot{\mathbf{q}}_y)\dot{\mathbf{q}}_y + \mathbf{D}(\dot{\mathbf{q}}_y) + \mathbf{G}(\mathbf{q}_y) = \mathbf{Q}_y, \tag{8}$$

where the state vector $\mathbf{q}_y$, mass matrix $\mathbf{M}(\mathbf{q}_y)$, gravity vector $\mathbf{G}(\mathbf{q}_y)$, friction vector $\mathbf{D}(\dot{\mathbf{q}}_y)$, vector of control torque along the y-axis $\mathbf{Q}_y$, and Coriolis-centrifugal vector $\mathbf{C}(\mathbf{q}_y, \dot{\mathbf{q}}_y)$ are, respectively, expressed as

$$\mathbf{q}_y = [x_k \;\; \theta_y]^T,$$

$$\mathbf{M}(\mathbf{q}_y) = \begin{bmatrix} m_k + \dfrac{I_k}{r_k^2} + \dfrac{3I_w\cos^2\alpha}{2r_w^2} + m_a & m_a l\cos\theta_y - \dfrac{3I_w\cos^2\alpha}{2r_w^2}r_k \\ m_a l\cos\theta_y - \dfrac{3I_w\cos^2\alpha}{2r_w^2}r_k & \dfrac{3I_w r_k^2\cos^2\alpha}{2r_w^2} + m_a l^2 + I_y \end{bmatrix},$$

ENERGY OF THE BALL

The kinetic energy of the ball is defined as the sum of the translational and rotational kinetic energies, that is,

$$T_{kx} = \frac{1}{2}\left(m_k + \frac{I_k}{r_k^2}\right)\dot{y}_k^2,$$

where m_k is the mass of the ball, I_k is the moment of inertia of the ball, and r_k is the ball radius. The potential energy of the ball is zero because the ball is assumed to only move on a horizontal surface, so $V_{kx} = 0$.

ENERGY OF THE BODY

The kinetic energy of the body is

$$T_{ax} = \frac{1}{2}I_x\dot{\theta}_x^2 + \frac{1}{2}m_a(\dot{y}_k - l\dot{\theta}_x\cos\theta_x)^2 + \frac{1}{2}m_a l^2\dot{\theta}_x^2\sin^2\theta_x,$$

where m_a is the body mass, I_x is the moment of inertia of the body around the x-axis, and l is the distance between the center of mass of the body and the ball. The potential energy of the robot body is

$$V_{ax} = m_a g l\cos\theta_x,$$

where g is the gravity acceleration.

ENERGY OF THE OMNIDIRECTIONAL WHEELS

The omnidirectional wheels are attached to the body. Thus, only the rotational motion of the omnidirectional wheels should be calculated. The kinetic and potential energies of the omnidirectional wheels are

$$T_{wx} = \frac{3I_w\cos^2\alpha}{4r_w^2}(\dot{y}_k + r_k\dot{\theta}_x)^2, V_{wx} = 0,$$

where I_w is the moment of inertia of the omnidirectional wheels.

The total kinetic energy T_x and the total potential energy V_x of the entire ball segway system are expressed as

$$\begin{aligned} T_x &= T_{kx} + T_{ax} + T_{wx} \\ &= \begin{pmatrix} \frac{1}{2}\left(m_k + \frac{I_k}{r_k^2}\right)\dot{y}_k^2 + \frac{1}{2}I_x\dot{\theta}_x^2 + \frac{1}{2}m_a(\dot{y}_k - l\dot{\theta}_x\cos\theta_x)^2 \\ + \frac{1}{2}m_a l^2\dot{\theta}_x^2\sin^2\theta_x + \frac{3I_w\cos^2\alpha}{4r_w^2}(\dot{y}_k + r_k\dot{\theta}_x)^2 \end{pmatrix}, \end{aligned}$$

and

$$V_x = V_{kx} + V_{ax} + V_{wx} = m_a g l\cos\theta_x,$$

respectively. The Lagrangian function L_x is $L_x = T_x - V_x$. The relationship between the torque input τ_x and the torques that directly control the minimal coordinates $\mathbf{q}_x$ is

$$\mathbf{Q}_x = \begin{bmatrix} \tau_{yk} \\ \tau_{\theta_x} \end{bmatrix} = \frac{1}{r_w}\begin{bmatrix} 1 \\ r_k \end{bmatrix}\tau_x.$$

Moreover, the friction vector is defined as

$$\mathbf{D}(\dot{\mathbf{q}}_x) = [b_y\dot{y}_k + d_y\text{sign}(\dot{y}_k) \;\; b_{rx}\dot{\theta}_x + d_{rx}\text{sign}(\dot{\theta}_x)]^T,$$

where b_y and b_{rx} are the viscous damping coefficients that model the spherical wheel–ground friction and spherical wheel–body friction in the y–z plane, respectively, and d_y and d_{rx} are the Coulomb friction coefficients that model the spherical wheel–ground and spherical wheel–body friction in the y–z plane, respectively.

The dynamic equations of the planar motion in the y–z plane can be obtained using the Euler–Lagrange equation and rewritten in matrix form

$$\mathbf{M}(\mathbf{q}_x)\ddot{\mathbf{q}}_x + \mathbf{C}(\mathbf{q}_x, \dot{\mathbf{q}}_x)\dot{\mathbf{q}}_x + \mathbf{D}(\dot{\mathbf{q}}_x) + \mathbf{G}(\mathbf{q}_x) = \mathbf{Q}_x,$$

where $\mathbf{M}(\mathbf{q}_x)$ is the mass matrix, $\mathbf{C}(\mathbf{q}_x, \dot{\mathbf{q}}_x)$ is the Coriolis-centrifugal vector, and $\mathbf{G}(\mathbf{q}_x)$ is the gravity vector.

The mass matrix is expressed as

$$\mathbf{M}(\mathbf{q}_x) = \begin{bmatrix} m_k + \frac{I_k}{r_k^2} + m_a + \frac{3I_w\cos^2\alpha}{2r_w^2} & \frac{3I_w\cos^2\alpha}{2r_w^2}r_k - m_a l\cos\theta_x \\ \frac{3I_w\cos^2\alpha}{2r_w^2}r_k - m_a l\cos\theta_x & m_a l^2 + \frac{3I_w r_k^2\cos^2\alpha}{2r_w^2} + I_x \end{bmatrix}.$$

The Coriolis-centrifugal vector is expressed as

$$\mathbf{C}(\mathbf{q}_x, \dot{\mathbf{q}}_x) = \begin{bmatrix} 0 & m_a l\dot{\theta}_x\sin\theta_x \\ 0 & 0 \end{bmatrix}.$$

The vector of gravitational forces is

$$\mathbf{G}(\mathbf{q}_x) = [0 \;\; -m_a g l\sin\theta_x]^T.$$

$$\begin{aligned} \mathbf{G}(\mathbf{q}_y) &= [0 \;\; -m_a g l\sin\theta_y]^T, \\ \mathbf{D}(\dot{\mathbf{q}}_y) &= [b_x\dot{x}_k + d_x sign(\dot{x}_k) \;\; b_{ry}\dot{\theta}_y + d_{ry} sign(\dot{\theta}_y)]^T, \\ \mathbf{Q}_y &= \left[-\frac{1}{r_w}\tau_y \;\; \frac{r_k}{r_w}\tau_y\right]^T, \\ \mathbf{C}(\mathbf{q}_y, \dot{\mathbf{q}}_y) &= \begin{bmatrix} 0 & -m_a l\dot{\theta}_y\sin\theta_y \\ 0 & 0 \end{bmatrix}, \end{aligned}$$

where τ_y is the equivalent torque of three driving motors acting on the ball around the y-axis; b_x and b_{ry} are the viscous damping coefficients that model the spherical wheel–ground friction and spherical wheel–body friction in the x–z plane, respectively; and d_x and d_{ry} are the Coulomb friction coefficients that model the spherical wheel–ground friction and spherical wheel–body friction in the x–z plane, respectively.

State vectors $\mathbf{x}_r = [\mathbf{q}_x^T \;\; \dot{\mathbf{q}}_x^T]^T = [y_k \;\; \theta_x \;\; \dot{y}_k \;\; \dot{\theta}_x]^T$ and $\mathbf{x}_p = [\mathbf{q}_y^T \;\; \dot{\mathbf{q}}_y^T]^T = [x_k \;\; \theta_y \;\; \dot{x}_k \;\; \dot{\theta}_y]^T$, and control inputs $u_x = \tau_x$ and $u_y = \tau_y$ are defined to convert the dynamic equations in the two vertical planes into standard nonlinear state-space forms and combine them with (7) or (8), which yields

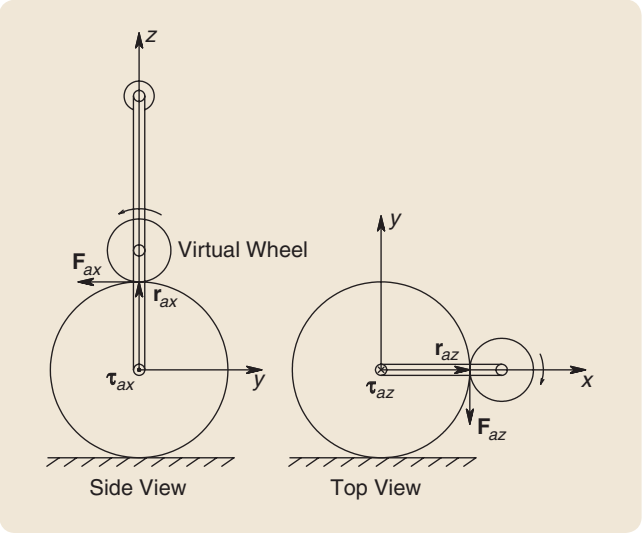


FIGURE 5 A sketch of the body torque generated by the virtual motor torques, tangential forces, and corresponding levers. The omnidirectional wheel and driving motors are modeled as virtual actuating wheels. Each two-dimensional (2-D) model contains only one virtual actuating wheel, which rotates around the shaft orthogonal to the plane of the 2-D model. The tangential force $\mathbf{F}_{az}, \mathbf{F}_{ay}$, and $\mathbf{F}_{az}$ are functions of the equivalent torques $\boldsymbol{\tau}_{ax}, \boldsymbol{\tau}_{ay}$, and $\boldsymbol{\tau}_{az}$ around the x-, y-, and z-axes and the levers $\mathbf{r}_{ax}, \mathbf{r}_{ay}$, and $\mathbf{r}_{az}$.

$$\dot{\mathbf{x}}_r = \begin{bmatrix} \dot{\mathbf{q}}_x \\ \mathbf{M}^{-1}(\mathbf{q}_x)[\mathbf{Q}(u_x) - \mathbf{C}(\mathbf{q}_x, \dot{\mathbf{q}}_x)\dot{\mathbf{q}}_x - \mathbf{D}(\dot{\mathbf{q}}_x) - \mathbf{G}(\mathbf{q}_x)] \end{bmatrix} = \mathbf{F}_r(\mathbf{x}_r, u_x), \tag{9}$$

$$\dot{\mathbf{x}}_p = \begin{bmatrix} \dot{\mathbf{q}}_y \\ \mathbf{M}^{-1}(\mathbf{q}_y)[\mathbf{Q}(u_y) - \mathbf{C}(\mathbf{q}_y, \dot{\mathbf{q}}_y)\dot{\mathbf{q}}_y - \mathbf{D}(\dot{\mathbf{q}}_y) - \mathbf{G}(\mathbf{q}_y)] \end{bmatrix} = \mathbf{F}_p(\mathbf{x}_p, u_y), \tag{10}$$

where

$$\mathbf{F}_r(\mathbf{x}_r, u_x) = \begin{bmatrix} \dot{y}_k \\ \dot{\theta}_x \\ f_{r3}(\mathbf{x}_r) + g_{r3}(\mathbf{x}_r)u_x \\ f_{r4}(\mathbf{x}_r) + g_{r4}(\mathbf{x}_r)u_x \end{bmatrix}, \tag{11}$$

$$\mathbf{F}_p(\mathbf{x}_p, u_y) = \begin{bmatrix} \dot{x}_k \\ \dot{\theta}_y \\ f_{p3}(\mathbf{x}_p) + g_{p3}(\mathbf{x}_p)u_y \\ f_{p4}(\mathbf{x}_p) + g_{p4}(\mathbf{x}_p)u_y \end{bmatrix}. \tag{12}$$

Using the ball segway system parameters in Table 1 yields

$$f_{r3}(\mathbf{x}_r) = -0.05\begin{pmatrix} \dfrac{7.59\sin\theta_x - 65.5\sin 2\theta_x + 11.5\dot{\theta}_x^2\sin\theta_x}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2} \\ + \dfrac{0.45b_y\dot{y}_k - 0.03b_{rx}\dot{\theta}_x + 0.52b_{rx}\dot{\theta}_x\cos\theta_x}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2} \\ + \dfrac{0.45d_y sign(\dot{y}_k) - 0.03d_{rx}sign(\dot{\theta}_x)}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2} \\ + \dfrac{0.52d_{rx}sign(\dot{\theta}_x)\cos\theta_x}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2} \end{pmatrix},$$

$$g_{r3}(\mathbf{x}_r) = \frac{0.06\cos\theta_x + 0.21}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2},$$

$$f_{r4}(\mathbf{x}_r) = \begin{pmatrix} \dfrac{25.1\sin\theta_x + 0.037\dot{\theta}_x^2\sin\theta_x - 0.32\theta_x^2\sin 2\theta_x}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2} \\ - \dfrac{0.1b_{rx}\dot{\theta}_x - 0.0015b_y\dot{y}_k + 0.025b_y\dot{y}_k\cos\theta_x}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2} \\ - \dfrac{0.1d_{rx}sign(\dot{\theta}_x) - 0.0015d_y sign(\dot{y}_k)}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2} \\ + \dfrac{0.025d_y sign(\dot{y}_k)\cos\theta_x}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2} \end{pmatrix},$$

$$g_{r4}(\mathbf{x}_r) = \frac{0.25\cos\theta_x + 0.21}{0.075\cos\theta_x - 0.65\cos^2\theta_x + 2.2};$$

$$f_{p3}(\mathbf{x}_p) = -0.05\begin{pmatrix} \dfrac{65.5\sin 2\theta_y - 7.59\sin\theta_y - 11.3\dot{\theta}_y^2\sin\theta_y}{0.075\cos\theta_y - 0.65\cos\theta_y + 2.2} \\ + \dfrac{0.03b_{ry}\dot{\theta}_y + 0.44b_x\dot{y}_k - 0.52b_{ry}\dot{\theta}_y\cos\theta_y}{0.075\cos\theta_y - 0.65\cos\theta_y + 2.2} \\ + \dfrac{0.03d_{ry}sign(\dot{\theta}_y) + 0.44d_x sign(\dot{x}_k)}{0.075\cos\theta_y - 0.65\cos\theta_y + 2.2} \\ - \dfrac{0.52d_{ry}sign(\dot{\theta}_y)\cos\theta_y}{0.075\cos\theta_y - 0.65\cos\theta_y + 2.2} \end{pmatrix},$$

$$g_{p3}(\mathbf{x}_p) = -\frac{0.06\cos\theta_y + 0.21}{0.075\cos\theta_y - 0.65\cos^2\theta_y + 2.2},$$

$$f_{p4}(\mathbf{x}_p) = -\begin{pmatrix} \dfrac{-25.1\sin\theta_y - 0.037\dot{\theta}_y^2\sin\theta_y + 0.32\theta_y^2\sin 2\theta_y}{0.075\cos\theta_y - 0.65\cos^2\theta_y + 2.2} \\ + \dfrac{0.1b_{ry}\dot{\theta}_y + 0.0015b_x\dot{x}_k - 0.025b_x\dot{x}_k\cos\theta_y}{0.075\cos\theta_y - 0.65\cos^2\theta_y + 2.2} \\ + \dfrac{0.1d_{ry}sign(\dot{\theta}_y) + 0.0015d_x sign(\dot{x}_k)}{0.075\cos\theta_y - 0.65\cos^2\theta_y + 2.2} \\ - \dfrac{0.025d_x sign(\dot{x}_k)\cos\theta_y}{0.075\cos\theta_y - 0.65\cos^2\theta_y + 2.2} \end{pmatrix},$$

$$g_{p4}(\mathbf{x}_p) = \frac{0.25\cos\theta_y + 0.21}{0.075\cos\theta_y - 0.65\cos^2\theta_y + 2.2}.$$

The dynamic equation obtained in the x–y plane is

$$\ddot{\theta}_z = \frac{I_k}{I_k I_z + 3(I_k + I_z)I_w\dfrac{r_k^2}{r_w^2}\sin^2\alpha}\frac{r_k}{r_w}\tau_z, \tag{13}$$

where τ_z is the equivalent torque of three driving motors that act on the ball around the z-axis.

Torque Conversion

In this section, a relationship between $\boldsymbol{\tau}_{aj}$ [the torque vector generated by $\boldsymbol{\tau}_j$ along the direction of the j-axis with $j = x, y, z$ (virtual drive system as shown in Figure 5)] and $\boldsymbol{\tau}_{ai}$ [the torque vector generated by omnidirectional wheel i with $i = 1, 2, 3$ (actual drive system as indicated in Figure 6)] is derived. This relationship can be obtained because the resulting torque of the body is conserved as [23]

$$\boldsymbol{\tau}_{ax} + \boldsymbol{\tau}_{ay} + \boldsymbol{\tau}_{az} = \boldsymbol{\tau}_{a1} + \boldsymbol{\tau}_{a2} + \boldsymbol{\tau}_{a3}. \tag{14}$$

$\mathbf{F}_{ai}$ is the tangential force acting on the ball orthogonal to $\boldsymbol{\tau}_{ai}$, which is generated by the torque of the ith omnidirectional

wheel, and $\mathbf{r}_{ai}$ is the lever corresponding to $\mathbf{F}_{ai}$. In addition, $\mathbf{F}_{aj}$ is the tangential force acting on the ball orthogonal to $\boldsymbol{\tau}_{aj}$, which is generated by torques in the direction of the j-axis with $j = x, y, z$, and $\mathbf{r}_{aj}$ is the lever corresponding to $\mathbf{F}_{aj}$. The torque vectors can be calculated as

$$\begin{aligned}\boldsymbol{\tau}_{ai} &= \mathbf{F}_{ai} \times \mathbf{r}_{ai},\ i = 1,2,3,\\ \boldsymbol{\tau}_{aj} &= \mathbf{F}_{aj} \times \mathbf{r}_{aj},\ j = x,y,z.\end{aligned} \tag{15}$$

Based on Figure 6, the following tangential forces of the actual drive system are derived as

$$\begin{aligned}\mathbf{F}_{a1} &= \frac{\tau_1}{r_w}[0\ \ -1\ \ 0]^T,\\ \mathbf{F}_{a2} &= \frac{\tau_2}{r_w}\left[\frac{\sqrt{3}}{2}\ \ \frac{1}{2}\ \ 0\right]^T,\\ \mathbf{F}_{a3} &= \frac{\tau_3}{r_w}\left[-\frac{\sqrt{3}}{2}\ \ \frac{1}{2}\ \ 0\right]^T.\end{aligned} \tag{16}$$

The levers of the actual drive system are defined by

$$\begin{aligned}\mathbf{r}_{a1} &= r_k[\sin\alpha\ \ 0\ \ \cos\alpha]^T,\\ \mathbf{r}_{a2} &= r_k\left[-\frac{1}{2}\sin\alpha\ \ \frac{\sqrt{3}}{2}\sin\alpha\ \ \cos\alpha\right]^T,\\ \mathbf{r}_{a3} &= r_k\left[-\frac{1}{2}\sin\alpha\ \ -\frac{\sqrt{3}}{2}\sin\alpha\ \ \cos\alpha\right]^T.\end{aligned} \tag{17}$$

In addition, based on Figure 5, the following tangential forces of the virtual drive system are obtained as

$$\begin{aligned}\mathbf{F}_{ax} &= \frac{\tau_x}{r_w}[0\ \ -1\ \ 0]^T,\\ \mathbf{F}_{ay} &= \frac{\tau_y}{r_w}[1\ \ 0\ \ 0]^T,\\ \mathbf{F}_{az} &= \frac{\tau_z}{r_w}[0\ \ -1\ \ 0]^T.\end{aligned} \tag{18}$$

The levers of the virtual drive system are defined by

$$\begin{aligned}\mathbf{r}_{ax} = \mathbf{r}_{ay} &= r_k[0\ \ 0\ \ 1]^T,\\ \mathbf{r}_{az} &= \mathbf{r}_k[1\ \ 0\ \ 0]^T.\end{aligned} \tag{19}$$

Substituting (16), (19), and (15) into (14) and solving (14) for the actual motor torques τ_1, τ_2, and τ_3 as functions of the equivalent torques along the x-,y-, and z-axes, τ_x, τ_y, and τ_z in matrix form yield

$$\begin{bmatrix}\tau_1\\ \tau_2\\ \tau_3\end{bmatrix} = \begin{bmatrix}\frac{2}{3\cos\alpha} & 0 & \frac{1}{3\sin\alpha}\\ -\frac{1}{3\cos\alpha} & \frac{\sqrt{3}}{3\cos\alpha} & \frac{1}{3\sin\alpha}\\ -\frac{1}{3\cos\alpha} & -\frac{\sqrt{3}}{3\cos\alpha} & \frac{1}{3\sin\alpha}\end{bmatrix}\begin{bmatrix}\tau_x\\ \tau_y\\ \tau_z\end{bmatrix}. \tag{20}$$

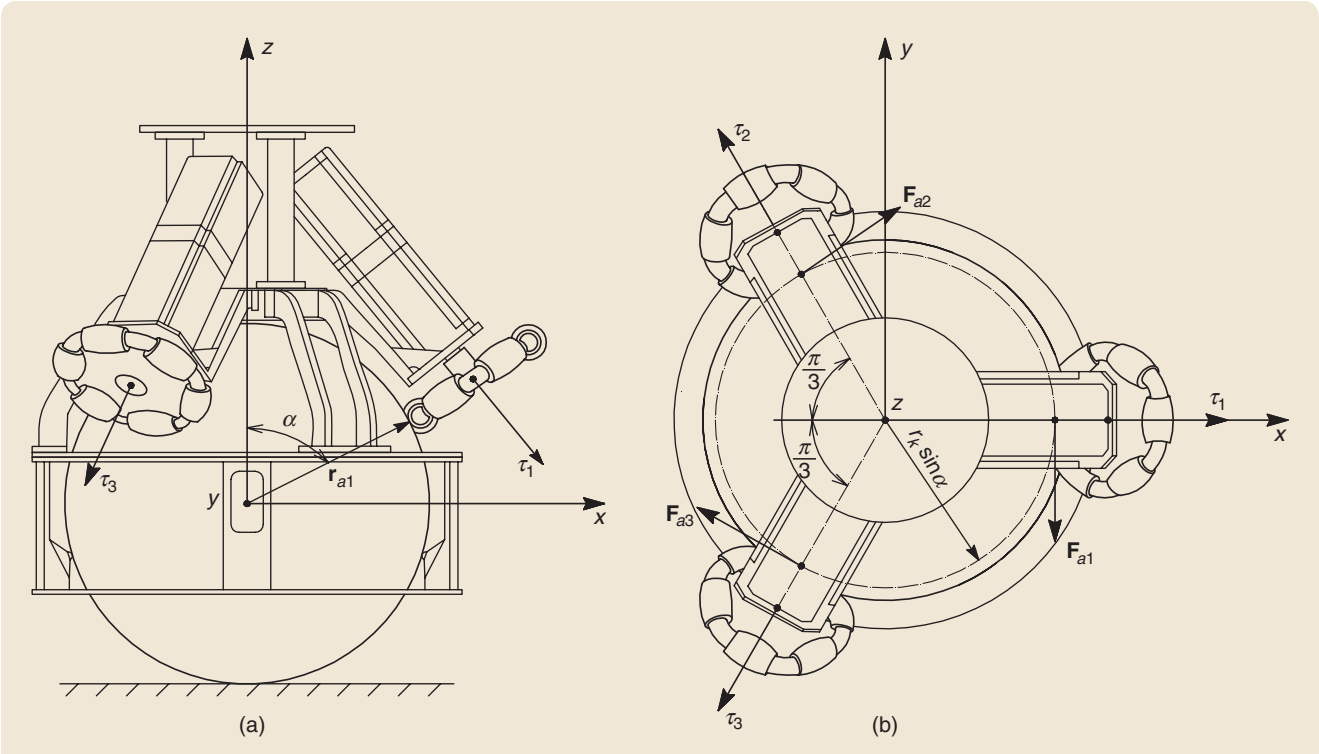


FIGURE 6 A sketch of body torque generated by the actual motor torques, tangential forces, and the corresponding levers. The tangential forces $\mathbf{F}_{a1}, \mathbf{F}_{a2}$, and $\mathbf{F}_{a3}$ are generated by torques of the driving motors $\boldsymbol{\tau}_1, \boldsymbol{\tau}_2$, and $\boldsymbol{\tau}_3$ in the actual configuration. The levers for this configuration are denoted as $\mathbf{r}_{a1}, \mathbf{r}_{a2}$, and $\mathbf{r}_{a3}$. (a) The torque of each omnidirectional wheel; the torque actuates the ball by scrolling on a circle on the ball, which is characterized by the motor arrangement zenith angle α. (b) A top view of the driving subsystem, where the torque of each omnidirectional wheel generates a tangential force on the surface of the ball with radius r_k.

Actuators

On the basis of torques conversion, τ_1, τ_2, and τ_3 represent the torques of the driving brushless dc motors. Characteristic equations of the brushless dc motor [24] are expressed as

$$L\frac{di}{dt} = -Ri - k_e\dot{\psi} + V, \tag{21}$$

$$I\frac{d^2\psi}{dt^2} = k_t i - k_f\dot{\psi} - \tau, \tag{22}$$

where k_e, k_t, and k_f are the back-electromotive force constant, torque constant, and linear viscous friction factor, respectively; R and L are the armature coil resistance and inductance, respectively; V and i denote the source voltage and armature current, respectively; τ is the torque load applied to the driving motor; and I represents the moment of inertia of the motor and load referred to the motor shaft.

For simplicity, the armature coil resistance and the moment of inertia of the motor are assumed to be comparably small and negligible. The dynamics of the driving motor can be obtained as

$$\tau = \frac{k_t}{R}V + \left(k_f - \frac{k_t k_e}{R}\right)\dot{\psi}. \tag{23}$$

Then, the generalized torques as functions of the input voltage and angular rate of the omnidirectional wheel are

$$\tau_1 = \frac{k_t}{R}V_1 + \left(k_f - \frac{k_t k_e}{R}\right)\dot{\psi}_1, \tag{24}$$

$$\tau_2 = \frac{k_t}{R}V_2 + \left(k_f - \frac{k_t k_e}{R}\right)\dot{\psi}_2, \tag{25}$$

$$\tau_3 = \frac{k_t}{R}V_3 + \left(k_f - \frac{k_t k_e}{R}\right)\dot{\psi}_3. \tag{26}$$

DESIGN OF THE CONTROL SYSTEM

This section describes the two controllers used for the ball segway. A double-loop controller, which consists of an LQR outer-loop controller and a PI-plus-feedforward inner-loop controller, is designed to balance and transfer the ball segway. The LQR and PI double-loop controller is an optimal controller that feeds all state variables to produce the optimal response. A remaining PD controller controls the robot that rotates around the vertical axis.

Control Architecture

This section presents the design of three improved double-loop controllers for self-balancing and station-keeping, robustness, and transferring control of the ball segway. Each controller has two loops, namely, a PI inner-loop controller and an LQR outer-loop controller. The three control targets, self-balancing, robustness, and transferring controls, are reduced to the same regulation problem, that is, θ_x and θ_y are stabilized at zero, and x_k and y_k reach the planned position set by x_{kd} and y_{kd}, respectively.

Although the controllers have the same structure, the functionalities of the PI inner-loop and LQR outer-loop controllers are different when they are applied to various tasks, including balancing, robustness, and transferring controls. The specific functionalities of the controllers are presented as follows.

Balancing Controller

The objective of the balancing controller is to balance the body with the desired angles, which are zero for a pure balancing operation. The LQR outer-loop controller provides the speed reference to the PI inner-loop controller. When the LQR and PI double-loop controller executes a balancing function, the speed reference of the PI inner-loop controller is maintained around zero by the LQR outer-loop controller. Meanwhile, the PI inner-loop controller maintains the ball segway in a vertical position and fixes it on the floor.

Station-keeping is an effort by the ball segway to retain a specific position on the floor using the LQR outer-loop controller.

Transferring Controller

The aforementioned balancing controller is good at balancing with the desired body angles; however, the ball segway cannot move to any desired ball position nor gain speed on the floor with the balancing controller. Therefore, a transferring controller is designed to control the movement of the ball segway.

Considering the transferring controller, the LQR outer-loop controller serves as a speed controller. The outputs of the LQR outer-loop controller are position and speed references, depending on the errors between the current and desired speeds of the ball and between the current and desired positions of the ball. The LQR outer-loop controller has two major applications. The first is a stopping controller that enables the ball segway to slow down and stop when subjected to large external disturbances. This application is utilized as a robust feature of the double-loop control system against external disturbances. The second is for movement where the user can provide velocity commands using a joystick. The control signal from the joystick is ramped to reduce the initial tilt angles of the body while controlling movement.

Robustness Controller

The PI inner-loop controller automatically compensates for various frictional torques for velocity tracking; thus, it reduces the effect of uncertainties of friction. In Figure 7, additional control torques u_{xf} and u_{yf} are considered as two feedforward controls that eliminate most uncertainties caused by viscous and Coulomb frictions as well as model and system parameters. The control gains of the PI inner-loop controller are empirically selected to provide proper performance for velocity tracking.

Rotating Controller

A PD controller is utilized to control the rotation of the robot around the vertical axis. The controller feeds back the

yaw angle θ_z and yaw angular velocity $\dot{\theta}_z$. The output of the rotational controller is the following torque–control input acting on the ball

$$\tau_z = K_{dz}\dot{\theta}_z + K_{pz}(\theta_z - \theta_z), \tag{27}$$

where K_{pz} and K_{dz} are the PD control gains of the rotational controller.

PI Plus Feedforward Inner-Loop Controller Design

The PI inner-loop controller, which feeds back the ball velocity, aims to control the ball position. The output of the inner-loop controller is the torque–control input acting on the ball segway, that is,

$$\begin{aligned} u_x &= u_{xf} + K_{pr}(v_y - \dot{y}_k) + K_{ir}(x_{5r} - y_k), \\ u_y &= u_{yf} + K_{pp}(v_x - \dot{x}_k) + K_{ip}(x_{5p} - x_k), \end{aligned} \tag{28}$$

where K_p and K_i are PI control gains; v_x and v_y are reference velocities of the ball along the x- and y-axes, respectively, and are also outputs of the outer-loop controller; x_5 is the reference position of the ball, with $\dot{x}_{5r} = v_y$, $\dot{x}_{5p} = v_x$; u_{xf} and u_{yf} are the feedforward compensation controls; and r and p denote roll and pitch, respectively.

The target of the PI inner-loop controller is to let $\dot{y}_k$ approach v_y as time approaches infinity. The velocity error is represented as

$$e_{vy}(t) = v_y - \dot{y}_k. \tag{29}$$

Differentiating $e_{vy}(t)$ with respect to time t yields

$$\dot{e}_{vy}(t) = \dot{v}_y - \ddot{y}_k = \dot{v}_y - [f_{r3}(\mathbf{x}_r) + g_{r3}(\mathbf{x}_r)u_x]. \tag{30}$$

Let

$$u_x = g_{r3}^{-1}(\mathbf{x}_r)\Big[-f_{r3}(\mathbf{x}_r) + \dot{v}_y + K'_{pr}e_{vy} + K'_{ir}\int_0^t e_{vy}(\tau)d\tau\Big]. \tag{31}$$

Substituting (31) into (30) yields

$$\dot{e}_{vy} = -K'_{pr}e_{vy} - K'_{ir}\int_0^t e_{vy}(\tau)d\tau. \tag{32}$$

Differentiating (32) with respect to time t gives

$$\ddot{e}_{vy} + K'_{pr}\dot{e}_{vy} + K'_{ir}e_{vy} = 0. \tag{33}$$

To stabilize the closed-loop system and obtain the desired transient performance simultaneously, let the closed-loop error (33) equal the desired standard second-order characteristic equation, that is,

$$s^2 + K'_{pr}s + K'_{ir} = s^2 + 2\xi\omega_n s + \omega_n^2, \tag{34}$$

where $\omega_n > 0$ is the desired natural frequency, and $\xi > 0$ is the desired damping ratio. Comparing the coefficients of polynomials on two sides yields

$$K'_{pr} = 2\xi\omega_n, \qquad K'_{ir} = \omega_n^2. \tag{35}$$

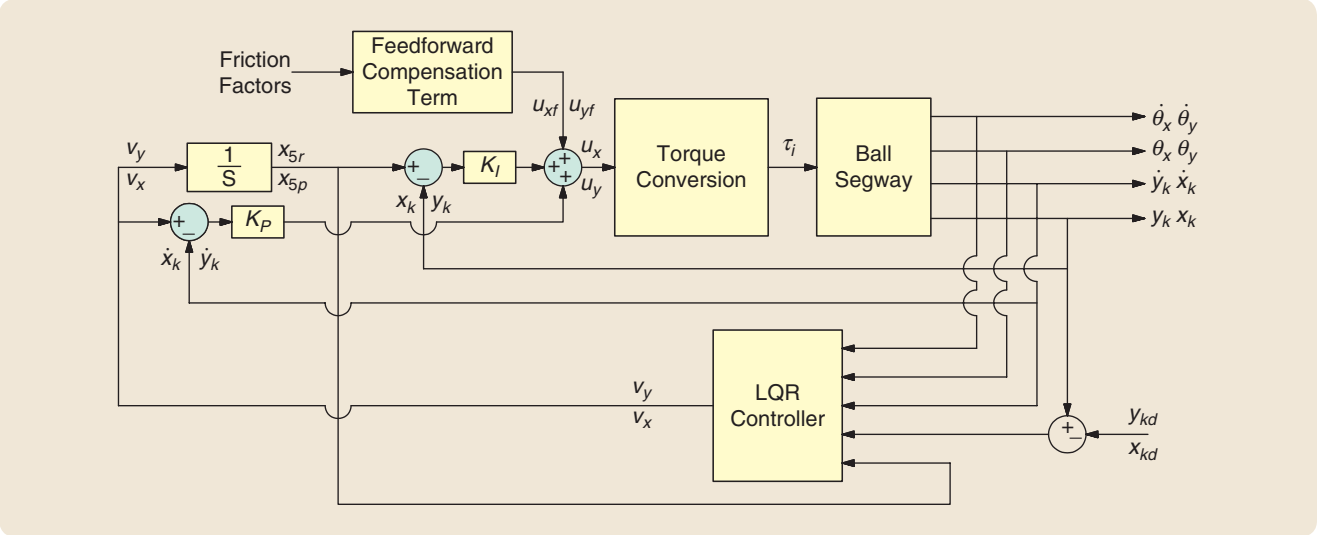


FIGURE 7 The architecture of the double-loop control system. This control diagram is employed to implement the control of the ball segway in numerical simulations and experiments. The control system includes a proportional-integral (PI) inner-loop controller and a linear-quadratic regulator (LQR) outer-loop controller. Based on the two-dimensional model of the ball segway, the double-loop control systems are designed to control balancing, station-keeping, and point-to-point transferring. State variables $\mathbf{x}_r = [y_k \;\; \theta_x \;\; \dot{y}_k \;\; \dot{\theta}_x]^T$ and $\mathbf{x}_p = [x_k \;\; \theta_y \;\; \dot{x}_k \;\; \dot{\theta}_y]^T$ feedback the LQR outer-loop controller. The outputs v_x and v_y of the outer-loop controller are reference speeds for the PI inner-loop controller. The required torque of each driving motor $\tau_i (i = 1, 2, 3)$ is given as a function of torque inputs u_x and u_y using the torque conversion. Additional control torques u_{xf} and u_{yf} are considered as two feedforward controls that eliminate the uncertainties of friction, model, and external disturbances. y_{kd} and x_{kd} represent the reference position of the ball.

Substituting (35) into (31) yields

$$u_x = -g_{r3}^{-1}(\mathbf{x}_r) f_{r3}(\mathbf{x}_r) + g_{r3}^{-1}(\mathbf{x}_r)\dot{v}_y + 2g_{r3}^{-1}(\mathbf{x}_r)\xi\omega_n e_{vy} + g_{r3}^{-1}(\mathbf{x}_r)\omega_n^2 \int_0^t e_{vy}(\tau)d\tau. \quad (36)$$

In the linearized case of an equilibrium point, $\dot{v}_y = 0$, $f_{r3}(\mathbf{x}_r) = f_{r3}(\mathbf{0})$, and $g_{r3}(\mathbf{x}_r) = g_{r3}(\mathbf{0})$. Therefore, the control input u_x are expressed as

$$u_x = -g_{r3}^{-1}(\mathbf{0}) f_{r3}(\mathbf{0}) + 2g_{r3}^{-1}(\mathbf{0})\xi\omega_n e_{vy} + g_{r3}^{-1}(\mathbf{0})\omega_n^2 \int_0^t e_{vy}(\tau)d\tau = u_{xf} + K_{pr} e_{vy} + K_{ir} \int_0^t e_{vy}(\tau)d\tau, \quad (37)$$

where

$$K_{pr} = 2g_{r3}^{-1}(\mathbf{0})\xi\omega_n, \quad (38)$$
$$K_{ir} = g_{r3}^{-1}(\mathbf{0})\omega_n^2, \quad (39)$$
$$u_{xf} = -g_{r3}^{-1}(\mathbf{0}) f_{r3}(\mathbf{0}). \quad (40)$$

The feedforward compensation control (40) eliminates most uncertainties of friction factors. Based on the aforementioned control design steps, the control gains K_{pp} and K_{ip}, and the feedforward control u_{yf} are completely determined as

$$K_{pp} = 2g_{p3}^{-1}(\mathbf{0})\xi\omega_n, \quad (41)$$
$$K_{ip} = g_{p3}^{-1}(\mathbf{0})\omega_n^2, \quad (42)$$
$$u_{yf} = -g_{p3}^{-1}(\mathbf{0}) f_{p3}(\mathbf{0}). \quad (43)$$

LQR Outer-loop Controller Design

In designing the LQR outer-loop controller, the integral term x_{5r} in the y–z plane or x_{5p} in the x–z plane is added into an extra state to the 2-D model system. The augmented state vectors are $\mathbf{x}_{ra} = [\mathbf{x}_r^T \;\; x_{5r}]^T$ and $\mathbf{x}_{pa} = [\mathbf{x}_p^T \;\; x_{5p}]^T$, and the ball segway dynamics with the PI inner-loop control law without feedforward compensation term in the y–z and x–z planes are written as

$$\dot{\mathbf{x}}_{ra} = \begin{bmatrix} \mathbf{F}_r(\mathbf{x}_r, K_{pr}(\nu_y - \dot{y}_k) + K_{ir}(x_{5r} - y_k)) \\ \nu_y \end{bmatrix} = \mathbf{F}_{ra}(\mathbf{x}_{ra}, \nu_y),$$
$$\dot{\mathbf{x}}_{pa} = \begin{bmatrix} \mathbf{F}_p(\mathbf{x}_p, K_{pp}(\nu_x - \dot{x}_k) + K_{ip}(x_{5p} - x_k)) \\ \nu_x \end{bmatrix} = \mathbf{F}_{pa}(\mathbf{x}_{pa}, \nu_x). \quad (44)$$

An LQR outer-loop controller is then designed by initially linearizing the robot dynamics (44). Based on the Taylor series expansion of an equilibrium point, where $\mathbf{x}_{ra} = \mathbf{0}$ and $\mathbf{x}_{pa} = \mathbf{0}$, the linearized dynamic model is

$$\dot{\mathbf{x}}_{ra} = \mathbf{A}_r \mathbf{x}_{ra} + \mathbf{B}_r \nu_y,$$
$$\dot{\mathbf{x}}_{pa} = \mathbf{A}_p \mathbf{x}_{pa} + \mathbf{B}_p \nu_x, \quad (45)$$

where

$$\mathbf{A}_r = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ a_{r31} & a_{r32} & a_{r33} & a_{r34} & a_{r35} \\ a_{r41} & a_{r42} & a_{r43} & a_{r44} & a_{r45} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$
$$\mathbf{B}_r = [0 \;\; 0 \;\; b_{r3} \;\; b_{r4} \;\; 1]^T,$$
$$\mathbf{A}_p = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ a_{p31} & a_{p32} & a_{p33} & a_{p34} & a_{p35} \\ a_{p41} & a_{p42} & a_{p43} & a_{p44} & a_{p45} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$
$$\mathbf{B}_p = [0 \;\; 0 \;\; b_{p3} \;\; b_{p4} \;\; 1]^T.$$

"Linear Model of the Two-Dimensional Ball Segway System" provides the elements of the matrices $\mathbf{A}r$, $\mathbf{B}r$, $\mathbf{A}p$, and $\mathbf{B}p$. By checking the rank of the controllability matrix [25], $\mathbf{C}_r = [\mathbf{B}_r \;\; \mathbf{A}_r\mathbf{B}_r \;\; \mathbf{A}_r^2\mathbf{B}_r \;\; \mathbf{A}_r^3\mathbf{B}_r \;\; \mathbf{A}_r^4\mathbf{B}_r]$ and $\mathbf{C}_p = [\mathbf{B}_p \;\; \mathbf{A}_p\mathbf{B}_p \;\; \mathbf{A}_p^2\mathbf{B}_p \;\; \mathbf{A}_p^3\mathbf{B}_p \;\; \mathbf{A}_p^4\mathbf{B}_p]$, thereby achieving a rating of five for $\mathbf{C}_r$ and $\mathbf{C}_p$. This result indicates that the planar systems with the PI inner-loop controller are controllable.

Given the system presented in (45), the optimal LQR outer-loop controller can be used to find a linear state-feedback controller that stabilizes all state variables of the ball segway system around the equilibrium point and minimizes the performance index, that is,

$$J_{ra} = \int_0^\infty (\mathbf{x}_{ra}^T \mathbf{Q}_{ra} \mathbf{x}_{ra} + R_{ra} \nu_y^2) dt,$$
$$J_{pa} = \int_0^\infty (\mathbf{x}_{pa}^T \mathbf{Q}_{pa} \mathbf{x}_{pa} + R_{pa} \nu_x^2) dt, \quad (46)$$

where $\mathbf{Q}_{ra}$ and $\mathbf{Q}_{pa}$ are symmetric and semidefinite positive matrices; and R_{ra} and R_{pa} are positive values.

The LQR problem can be viewed as the weighted minimization of a linear combination of the state $\mathbf{x}_{ra}$ and control input v_y as well as the state $\mathbf{x}_{pa}$ and control input v_x. The weighting matrices $\mathbf{Q}_{ra}$ and $\mathbf{Q}_{pa}$ establish which states should be controlled more tightly than the others. R_{ra} and R_{pa} weight the amount of control action to be applied to the performance index (46), depending on the size of state deviations. The structures of $\mathbf{Q}_{ra}$ and $\mathbf{Q}_{pa}$ are set as $\mathbf{Q}_{ra} = \text{diag}(\xi_{ry}, \xi_{r\theta}, \xi_{r\dot{y}}, \xi_{r\dot{\theta}}, \xi_{r5})$ and $\mathbf{Q}_{pa} = \text{diag}(\xi_{px}, \xi_{p\theta}, \xi_{p\dot{x}}, \xi_{p\dot{\theta}}, \xi_{p5})$, respectively, where ξ_{ry}, ξ_{px}, $\xi_{r\theta}$, $\xi_{p\theta}$, $\xi_{r\dot{y}}$, $\xi_{p\dot{x}}$, $\xi_{r\dot{\theta}}$, $\xi_{p\dot{\theta}}$, ξ_{r5}, and ξ_{p5} can be roughly considered as controlling the relative convergence rates of the ball position, body tilt angles, ball velocities, body angular velocities, and additional states, respectively.

The LQR outer-loop controller is obtained using the feedback control gain vectors $\mathbf{K}_r$ and $\mathbf{K}_p$. To calculate the optimal control gain vectors, the following equations are used:

$$\mathbf{K}_r = R_{ra}^{-1}\mathbf{B}_r\mathbf{P}_r, \mathbf{K}_p = R_{pa}^{-1}\mathbf{B}_p\mathbf{P}_p, \quad (47)$$
$$\mathbf{P}_r\mathbf{A}_r + \mathbf{A}_r^T\mathbf{P}_r - \mathbf{P}_r\mathbf{B}_r R_{ra}^{-1}\mathbf{B}_r^T\mathbf{P}_r + \mathbf{Q}_{ra} = \mathbf{0},$$
$$\mathbf{P}_p\mathbf{A}_p + \mathbf{A}_p^T\mathbf{P}_p - \mathbf{P}_p\mathbf{B}_p R_{pa}^{-1}\mathbf{B}_p^T\mathbf{P}_p + \mathbf{Q}_{pa} = \mathbf{0}. \quad (48)$$

Linear Model of the Two-Dimensional Ball Segway System

Consider

$$\dot{\mathbf{x}}_{ra} = \mathbf{F}_{ra}(x_{ra}, v_y),$$
$$\dot{\mathbf{x}}_{pa} = \mathbf{F}_{pa}(\mathbf{x}_{pa}, v_x).$$

$\mathbf{F}_{ra}(\mathbf{x}_{ra}, v_y)$ and $\mathbf{F}_{pa}(\mathbf{x}_{pa}, v_x)$ are assumed to be continuously differentiable. Based on the Taylor series expansion [S1] with the first-order term around an equilibrium point, where $\mathbf{x}_{ra} = \mathbf{0}$, $v_y = 0$ and $\mathbf{x}_{pa} = \mathbf{0}$, $v_x = 0$, the linear time-invariant equations are obtained as

$$\dot{\mathbf{x}}_{ra} = \mathbf{A}_r \mathbf{x}_{ra} + \mathbf{B}_r v_y,$$
$$\dot{\mathbf{x}}_{pa} = \mathbf{A}_p \mathbf{x}_{pa} + \mathbf{B}_p v_x,$$

where

$$\mathbf{A}_r = \frac{\partial \mathbf{F}_{ra}}{\partial \mathbf{x}_{ra}}\bigg|_{\mathbf{x}_{ra}=\mathbf{0}, v_y=0} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ a_{r31} & a_{r32} & a_{r33} & a_{r34} & a_{r35} \\ a_{r41} & a_{r42} & a_{r43} & a_{r44} & a_{r45} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$
$$\mathbf{B}_r = \frac{\partial \mathbf{F}_{ra}}{\partial \dot{y}_l}\bigg|_{\mathbf{x}_{ra}=\mathbf{0}, v_y=0} = [0 \;\; 0 \;\; b_{r3} \;\; b_{r4} \;\; 1]^T,$$
$$\mathbf{A}_p = \frac{\partial \mathbf{F}_{pa}}{\partial \mathbf{x}_{pa}}\bigg|_{\mathbf{x}_{pa}=\mathbf{0}, v_x=0} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ a_{p31} & a_{p32} & a_{p33} & a_{p34} & a_{p35} \\ a_{p41} & a_{p42} & a_{p43} & a_{p44} & a_{p45} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$
$$\mathbf{B}_p = \frac{\partial \mathbf{F}_{pa}}{\partial \dot{x}_l}\bigg|_{\mathbf{x}_{pa}=\mathbf{0}, v_x=0} = [0 \;\; 0 \;\; b_{p3} \;\; b_{p4} \;\; 1]^T.$$

The elements of the $\mathbf{A}_r, \mathbf{B}_r, \mathbf{A}_p$, and $\mathbf{B}_p$ matrices are

$$a_{r31} = -K_{ir} r_k^2 r_w (m_a l^2 + m_a r_k l + I_x) mau_r^{-1},$$
$$a_{r32} = m_a g l r_k^2 (2 m_a l r_w^2 - 3 I_w r_k \cos^2\alpha) mau_r^{-1},$$
$$a_{r33} = -r_k^2 [2(I_x + m_a l^2)(K_{pr} r_w + b_y r_w^2) + 3 I_w b_y r_k^2 \cos^2\alpha + 2 K_{pr} m_a l r_k r_w] mau_r^{-1},$$
$$a_{r34} = -b_{rx} r_k^2 (2 m_a l r_w^2 - 3 I_w r_k \cos^2\alpha) mau_r^{-1},$$
$$a_{r35} = K_{ir} r_k^2 r_w (m_a l^2 + m_a m_k l + I_x) mau_r^{-1},$$
$$a_{r41} = -2 K_{ir} r_k r_w [I_k + (m_a + m_k) r_k^2 + m_a l r_k] mau_r^{-1},$$
$$a_{r42} = m_a g l [2 I_k r_w^2 + 2(m_a + m_k) r_k^2 r_w^2 + 3 I_w r_k^2 \cos^2\alpha] mau_r^{-1},$$
$$a_{r43} = -r_k [2 K_{pr} r_w (I_k + m_a r_k^2 + m_k r_k^2 + m_a l r_k) - 3 I_w b_y r_k^2 \cos^2\alpha + 2 b_y m_a l r_k r_w^2] mau_r^{-1},$$
$$a_{r44} = -b_{rx} [2 I_k r_w^2 + 2(m_a + m_k) r_k^2 r_w^2 + 3 I_w r_k^2 \cos^2\alpha] mau_r^{-1},$$
$$a_{r45} = 2 K_{ir} r_k r_w [I_k + (m_a + m_k) r_k^2 + m_a l r_k] mau_r^{-1},$$
$$b_{r3} = 2 K_{pr} r_k^2 r_w (m_a l^2 + m_a l r_k + I_x) mau_r^{-1},$$
$$b_{r4} = 2 K_{pr} r_k r_w [I_k + (m_a + m_k) r_k^2 + m_a l r_k] mau_r^{-1},$$

where

$$mau_r = \left(2 r_w^2 I_k (I_x + m_a l^2) + 3 I_w (m_a + m_k) r_k^4 \cos^2\alpha + 2 I_x (m_a + m_k) r_k^2 r_w^2 + 3(I_k + I_x + 2 m_a l r_k) I_w r_k^2 \cos^2\alpha + 2 m_a m_k l^2 r_k^2 r_w^2 + 3 I_w m_a l^2 r_k^2 \cos^2\alpha\right),$$
$$a_{p31} = K_{ip} r_k^2 r_w (m_a l^2 + m_a r_k l + I_y) mau_p^{-1},$$
$$a_{p32} = -m_a g l r_k^2 (2 m_a l r_w^2 - 3 I_w r_k \cos^2\alpha) mau_p^{-1},$$
$$a_{r33} = 2 r_k^2 [I_y K_{pp} r_w - I_y b_x r_w^2 - 1.5 I_w b_x r_k^2 \cos^2\alpha + m_a l^2 r_w (K_{pp} - b_x r_w) + K_{pp} m_a l r_k r_w] mau_p^{-1},$$
$$a_{p34} = b_{ry} r_k^2 (2 m_a l r_w^2 - 3 I_w r_k \cos^2\alpha) mau_p^{-1},$$
$$a_{p35} = -K_{ip} r_k^2 r_w (m_a l^2 + m_a r_k l + I_y) mau_p^{-1},$$
$$a_{p41} = -2 K_{ip} r_k r_w [I_k + (m_a + m_k) r_k^2 + m_a l r_k] mau_p^{-1},$$
$$a_{p42} = m_a g l [2 I_k r_w^2 + 2(m_a + m_k) r_k^2 r_w^2 + 3 I_w r_k^2 \cos^2\alpha] mau_p^{-1},$$
$$a_{p43} = -r_k [2 K_{pp} r_w (I_k + m_a r_k^2 + m_k r_k^2 + m_a l r_k) + 3 I_w b_x r_k^2 \cos^2\alpha - 2 b_x m_a l r_k r_w^2] mau_p^{-1},$$
$$a_{p44} = -b_{ry} [2 I_k r_w^2 + 2(m_a + m_k) r_k^2 r_w^2 + 3 I_w r_k^2 \cos^2\alpha] mau_p^{-1},$$
$$a_{p45} = 2 K_{ip} r_k r_w [I_k + (m_a + m_k) r_k^2 + m_a l r_k] mau_p^{-1},$$
$$b_{p3} = -K_{pp} r_k^2 r_w (m_a l^2 + m_a l r_k + I_y) mau_p^{-1},$$
$$b_{p4} = 2 K_{pp} r_k r_w [I_k + (m_a + m_k) r_k^2 + m_a l r_k] mau_p^{-1},$$

where

$$mau_p = \left(2 r_w^2 I_k (I_y + m_a l^2) + 3 I_w (m_a + m_k) r_k^4 \cos^2\alpha + 2 I_y (m_a + m_k) r_k^2 r_w^2 + 3(I_k + I_y + 2 m_a l r_k) I_w r_k^2 \cos^2\alpha + 2 m_a m_k l^2 r_k^2 r_w^2 + 3 I_w m_a l^2 r_k^2 \cos^2\alpha\right).$$

REFERENCE

[S1] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control.* Englewood Cliffs, NJ: Prentice-Hall, 1991.

Equation (48) is the Riccati equation used to solve for the matrices $\mathbf{P}_p$ and $\mathbf{P}_r$, which are used to obtain the vectors $\mathbf{K}_r$ and $\mathbf{K}_p$. The optimal state feedback control law is then expressed as

$$v_y = -\mathbf{K}_r \mathbf{e}_{ra},$$
$$v_x = -\mathbf{K}_p \mathbf{e}_{pa}, \tag{49}$$

where $\mathbf{e}_{ra}$ and $\mathbf{e}_{pa}$ are the error vectors, and $\mathbf{K}_r$ and $\mathbf{K}_p$ are the optimal control gain vectors.

For the simulation test, the Matlab function *lqr* is used to calculate the associated gain vectors $\mathbf{K}_r$ and $\mathbf{K}_p$ with a given choice of $\mathbf{Q}_{ra}$, R_{ra} and $\mathbf{Q}_{pa}$, R_{pa}. In practice, however, the associated elements of these gain vectors are tuned by adjusting based on the simulation results because of the effect of noise and uncertainties of the robot parameters and model.

NUMERICAL SIMULATION

Initially, the control system applied to the ball segway is numerically simulated with the dynamics of the ball segway systems (7) and (8) and the linear control laws (28) and (49). The ball segway system parameters shown in Table 1 are simulated and experimentally tested. The design of the controller is conducted using Matlab function *lqr*. In Matlab and Simulink, all state variables of the robot system can be obtained from its mathematical model. For the following

numerical simulation, the control gains of the double-loop control system are fully tuned and selected.

- *PI inner-loop controller:* In the simulation, the PI inner-loop controller parameters can be obtained as $K_{pr} = K_{pp} = 6$, $K_{ir} = K_{ip} = 36$, with the desired natural frequency $\omega_n = 2.5$, desired damping ratio $\xi = 0.2$, and $g_{r3}(\mathbf{0}) = g_{p3}(\mathbf{0}) = 0.17$.
- *LQR outer-loop controller:* $\mathbf{Q}_{ra}, R_{ra}$ and $\mathbf{Q}_{pa}, R_{pa}$ are positive definite symmetric matrices, which is in the forms of $\mathbf{Q}_{ra} = \text{diag}(10, 2, 10, 2, 4)$, $R_{pa} = 4$ and $\mathbf{Q}_{pa} = \text{diag}(10, 5, 10, 5, 5)$, $R_{pa} = 2$. Moreover, the performance index (46) is optimized to obtain the associated state feedback results in vectors $\mathbf{K}_r$ and $\mathbf{K}_p$, which are expressed as $\mathbf{K}_r = [-6.98 \;\; -4.45 \;\; 20.88 \;\; 6.09 \;\; 5.11]$ and $\mathbf{K}_p = [-3.27 \;\; 4.01 \;\; 29.07 \;\; 8.54 \;\; 5.72]$.

The simulation is conducted with three tasks, including balancing control with an initial nonzero tilt angle, balancing control with external disturbances, and transferring control.

Balancing Control with an Initial Nonzero Tilt Angle

The first simulation investigates the balancing performances of the ball segway system. In this simulation, the ball segway is initially motionless and placed on the floor with a roll angle of 2° in the x-axis and a pitch angle of 2° in the y-axis from the upright position. The numerical results in Figure 8 show that the double-loop control scheme can stabilize overall state variables of the robot system. Thus, the PI-plus-feedforward inner-loop controller can overcome static friction.

Balancing Control with External Disturbances

For further validation of the robustness of the double-loop control system against external disturbances, an impulse disturbance force with an amplitude of 50 N is applied to the robot in the y–z plane, as well as in the x–z plane at 15 s after the balancing control begins.

The corresponding simulation responses are shown in Figure 9. When a kick is applied to the robot at 15 s, the tilt angles of the body in Figure 9(a) fluctuate the most at approximately 1.1° and then stabilize after 18 s. Figure 9(b) shows the ball trajectory on the floor. Under the action of the balancing controller, the robot continues to move in the direction of the kick to a new balancing position on the floor. The simulation results show that the ball segway system can stabilize all system state variables. This finding indicates that the double-loop controller can maintain balance with a large amount of external disturbances.

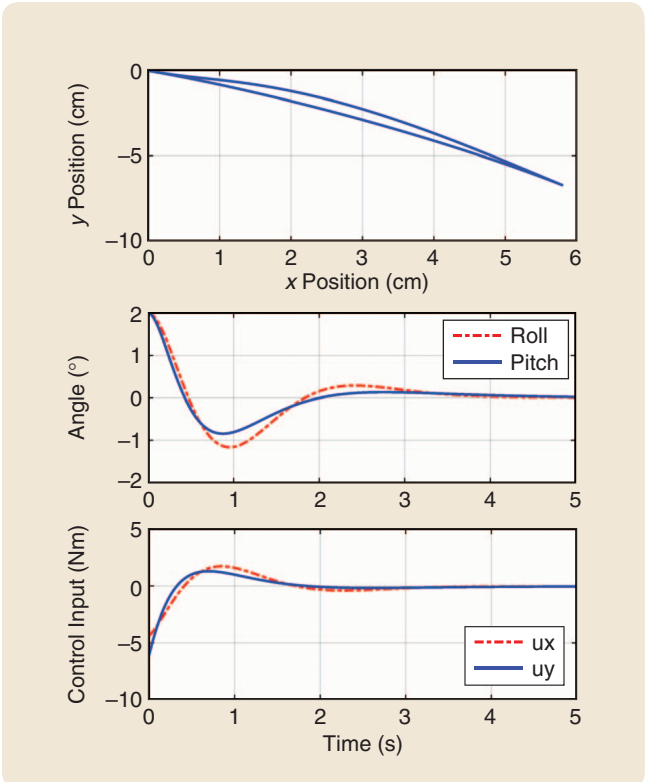


FIGURE 8 Simulation results for balancing control with an initial nonzero tilt angle. The nominal model corresponds to system parameters presented in Table 1. The control scheme can stabilize the system-state variables. The simulation results show that the controller is adequate in controlling balancing and station-keeping of the ball segway on the floor.

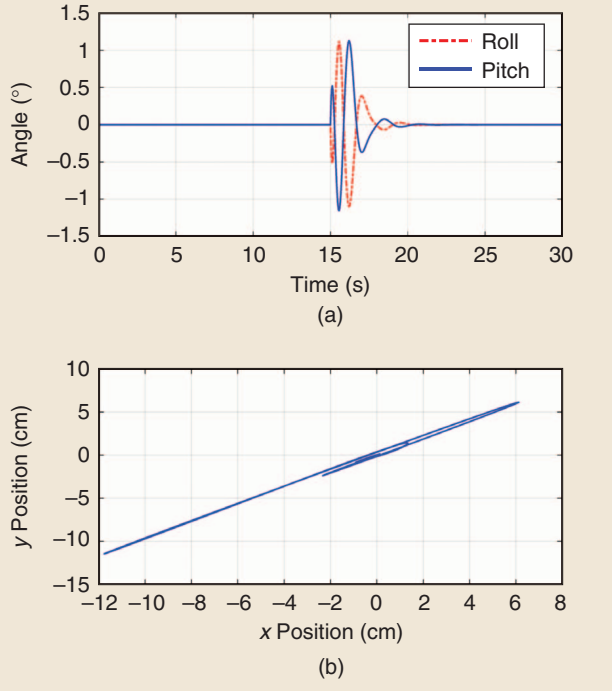


FIGURE 9 Simulation results of balancing control with external disturbances. The dotted line shows a response when a 50-N impulse is applied to the two-dimensional ball segway model in the x–z plane. The solid line shows the case when the same impulse is applied in the y–z plane. (a) The tilt angles of the body. (b) The trajectory of the ball on the floor.

Transferring Control

The transferring control results of the LQR and PI double-loop controller are presented to show the trajectory of the ball segway on the floor. For tracking control, the robot transfers from the initial position $(x_k, y_k) = (0, 0)$ to the goal position $(x_k, y_k) = (-0.13\text{ m}, 0.37\text{ m})$. The ball segway should ensure the balance of the body in the vertical position, following the model of an inverted pendulum when moving on the floor. The reference position should be ramped to create the tilt angle of the body with a small value. Figure 10 shows the point-to-point transferring control responses. The transferring simulation shows two instances of reference positions, step and ramp. The tilt angles of the body created when the reference position is a step are considerably larger than that of the body created when the reference position is a ramp. The simulation results show that the ramping position reference of the ball is necessary to control the transferring in point-to-point motion.

EXPERIMENT AND RESULT ANALYSIS

Ball Segway Hardware Structure

In this study, an Intel NUC minicomputer is used to acquire measurements and provide control input signals. Three circuit cards attached to the brushless dc motor blocks are used to send pulse width modulation signals that control the brushless dc motors and acquire signals from the encoders.

Three encoders with a resolution of 4000 counts/revolution and a microstrain IMU are used. The data from the three encoders, which are placed on the passive rollers of the driving motors, are used to estimate the position and velocity of the ball based on kinematics (5). The body angles are obtained by the IMU, which consists of an accelerometer with a resolution of <0.1 mg and a gyroscope with a resolution of <0.008°/s. The accelerometer measures linear accelerations of the robot along the x-, y-, and z-axes with high-frequency noise. The gyroscope measures the angular velocities around the x-, y-, and z-axes with low-frequency

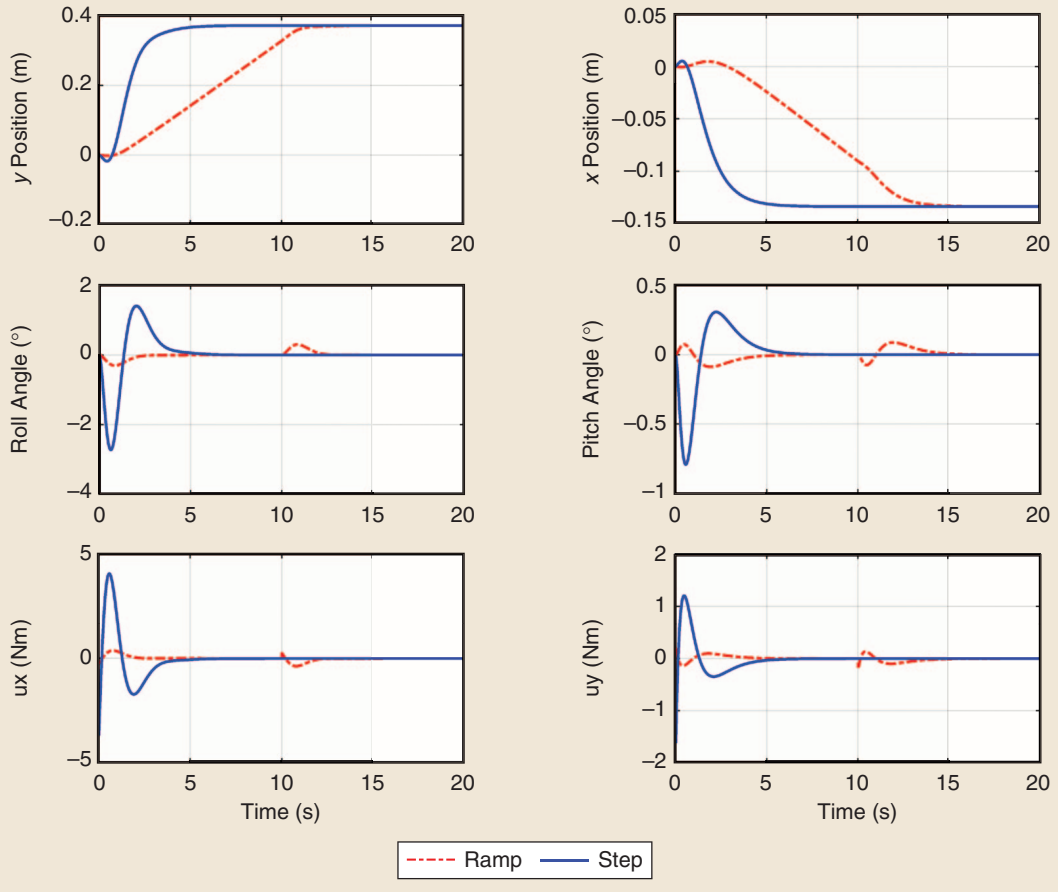
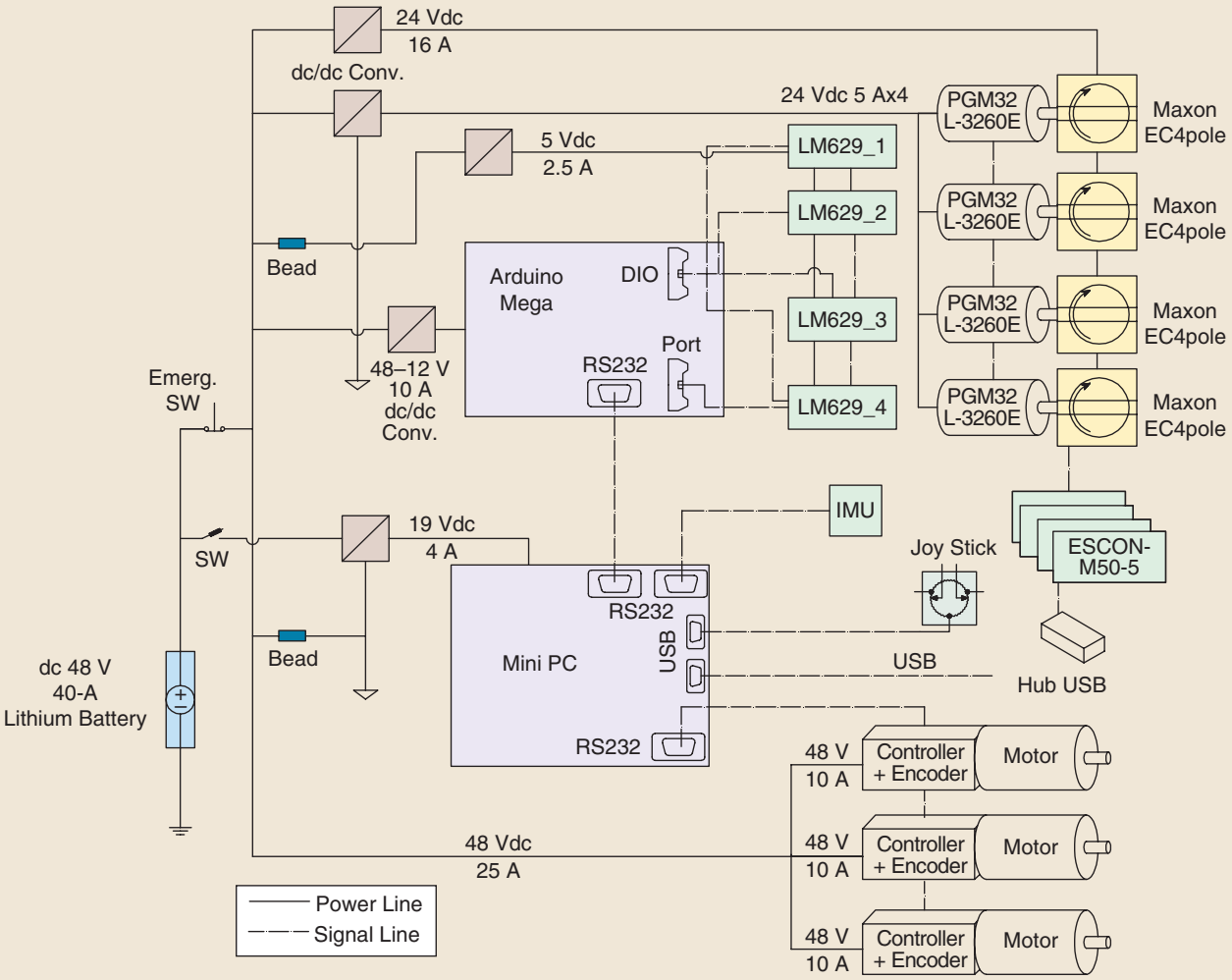
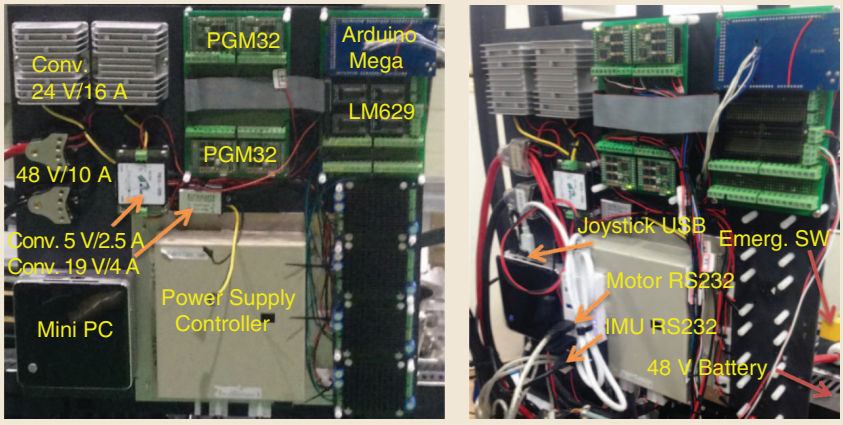


FIGURE 10 Simulation results of transferring control. The blue solid lines show the system responses when the desired reference input is given as a step position. The red dotted lines show the system responses when the desired reference input is given as a ramp, which is a velocity of 0.01 m/s. The controller can stabilize all state variables and control movement of the ball segway from the initial to the desired reference target.



(a)



(b)

FIGURE 11 A hardware block diagram and control board. The PC-based controller obtains tilt angles of the ball segway with an inertial measurement unit (IMU). Based on the obtained angle information, it compensates its posture by driving three Omni-Drive Modules. The control commands are transmitted to all Omni-Drive Modules, which are equipped with embedded motion controllers using RS-232 daisy chain communication interface. The other controller is an Arduino Mega 2560-based controller that interfaces four brushless dc controllers for speed control of the flywheel motor in each control moment gyroscope (CMG) and for rotation control of the gimbal motor in each CMG. (a) The hardware block diagram. (b) The controller system and electrical circuit.

noise. The IMU provides Kalman filtered pitch, roll, and yaw angles and angular rates with respect to gravity. The IMU and encoders provide all data required for full-state feedback control.

Two 48-V lithium battery packs supply power for three driving motors, a control moment gyroscope (CMG) cluster, sensors, and other devices with operating times of several hours. A wireless joystick is used to control the transferring of the ball segway in point-to-point motion.

The control algorithm is realized in the Intel NUC minicomputer using Visual Studio C++. The program also consists of torques conversion and kinematics to estimate the position of the robot on the floor. Its control interval is set at 15 ms. The control block diagram of the ball segway is shown in Figure 11.

The ball segway is controlled using the LQR and PI double-loop controller. The purpose of the experiment is to control the ball segway balancing and stable transferring. Four experimental cases, namely, balancing control with an initial zero tilt angle, balancing control with an initial nonzero tilt angle, balancing control with external disturbances, and transferring control, are illustrated in the succeeding subsections.

Balancing Control with an Initial Zero Tilt Angle

This section aims to present the performance of the double-loop control scheme for balancing and station-keeping. The balancing control at a fixed position is tested. Typical data obtained from the ball segway are shown in Figure 12. The robot is commanded to maintain its position while balancing on the floor. During balancing and station-keeping on the floor, the amount of deviation of the body tilt angles is at the most 0.4°, as shown in Figure 12(a). The angular velocities of the body change from –2°/s to 2°/s [Figure 12(b)]. Most trajectories of the ball segway during the experiment are within a rough ellipse with a major axis that has a length of 8.66 cm and a minor axis of 4.81 cm, as shown in Figure 12(c).

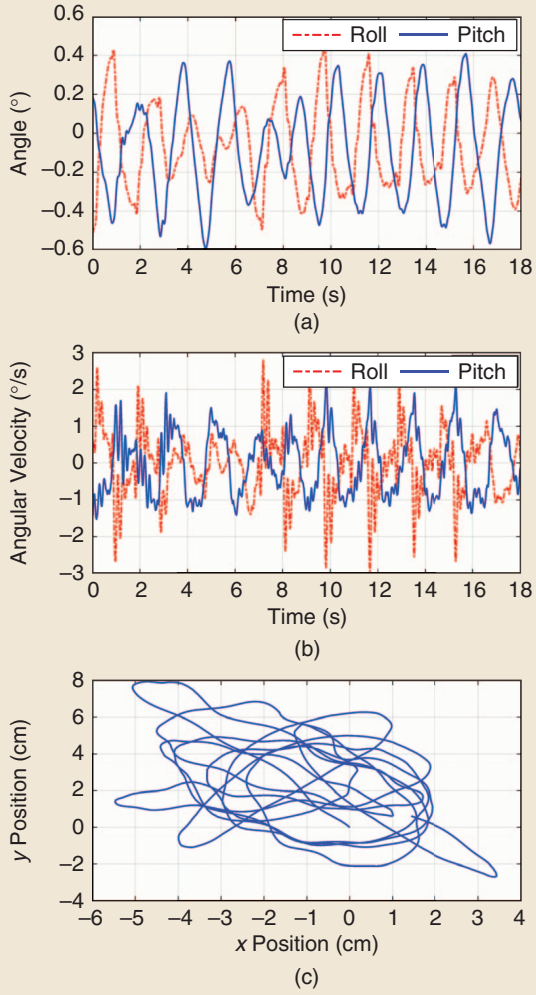


FIGURE 12 Experimental results of balancing control with an initial zero tilt angle. The controller is tuned to balance and station-keep the ball segway in a noisy environment. (a) Tilt angles of the body during balancing. (b) Angular velocities of the body during balancing. (c) The position of the ball on the floor.

Balancing Control with an Initial Nonzero Tilt Angle

Initially, the ball segway is out of power and placed on the floor with 5° tilt angles in the x- and y-axes from the upright position. Figure 13 shows the system responses while the ball segway stands itself from the initial power-off position by supplying power and controls balancing with the LQR and PI double-loop control algorithm. Figure 13(a) shows that the ball segway can balance at the vertical position after $t_s = 2$ s, where the steady-state error of the roll angle is 0.4° and that of the pitch angle is 0.3°. The ball segway then maintains balance at the new position at $(x_k, y_k) = (12.1\text{ cm}, -7.8\text{ cm})$ on the floor rather than staying at the origin, as indicated in Figure 13(c). The angular rates of the body are shown in Figure 13(b). The steady-state error of the roll rate is approximately 2.1°/s and that of the pitch rate is approximately 2.5°/s. Evidently, the double-loop controller can manage an initial nonzero tilt angle. The companion video [26] shows that the ball segway successfully balanced with an initial nonzero tilt angle.

Balancing Control with External Disturbances

Experiments are conducted to demonstrate the robust balancing stability of the ball segway against external disturbances. A control effort, which is considered as a feedforward control, is added to the system to eliminate friction, model uncertainties, and external disturbances. The experimental scenario is as follows: when the ball segway is balancing on the floor, a trained person is asked to kick it in a controlled manner; the measured value of the kick is approximately 350 N.

Figure 14 shows the responses of the ball segway in the kicking experiment. The trained personnel kicked the robot at 15.2 s after the experiment began, and the highest peak of the tilt angle of the body was obtained at that

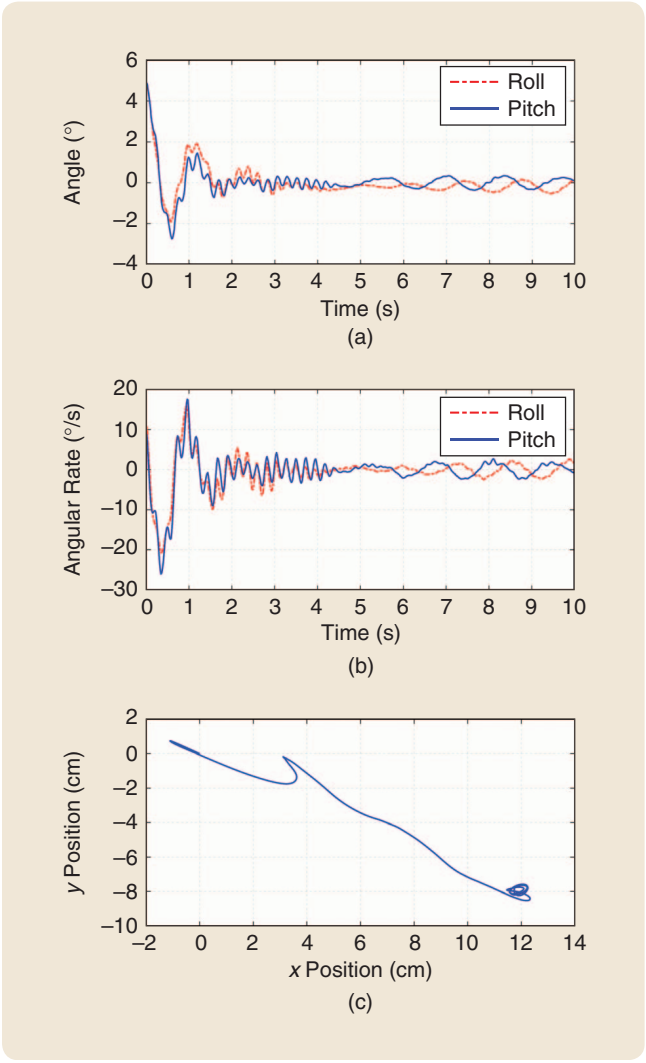


FIGURE 13 Experimental results of balancing control with an initial nonzero tilt angle. In this experiment, the ball segway is at the initial condition of 5° in the *x*-axis and 5° in the *y*-axis at a fixed position. (a) Roll and pitch angles of the body. (b) Roll and pitch rates of the body. (c) Position trajectories of the ball on the floor.

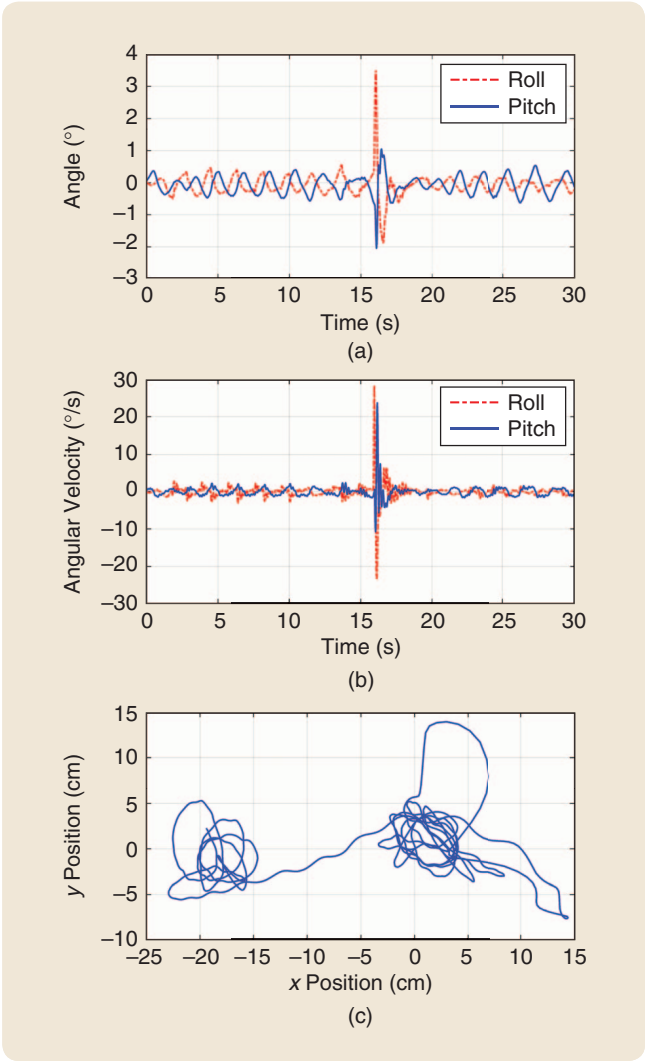


FIGURE 14 Experimental results of balancing control with large external disturbances. Under large disturbances, such as kicking, the ball segway can maintain balance and remain in a vertical position. However, under the action of the balancing controller, the ball segway continues to move in the direction of the force until the friction of the floor slows it down and brings it to rest at a new position on the floor. The experimental results prove the robustness of the double-loop control system when it is affected by large external disturbances. (a) Roll and pitch angles of the body during kicking. (b) Angular velocities of the body during kicking. (c) The position of the ball on the floor during kicking.

point, as shown in Figure 14(a). The corresponding trajectory of the ball segway in Figure 14(b) shows that it moved from its initial position to a new position around $(x_k, y_k) = (-18\,\text{cm}, -1\,\text{cm})$ on the floor and balanced at this new point. The experimental results indicate that the ball segway can efficiently resist a large amount of external disturbances. Obviously, the double-loop controller can manage large external disturbances, such as kicking. Some investigation results are shown in [26].

Transferring Control

The transferring experiment has encountered a problem related to the large mass and inertia of the ball and the entire system. Driving motors should create large torques to overcome static friction of the system with heavy weight and prevent large inertial force when stopping the ball segway. Several trials of various linear control methods of a single-loop control structure were unsuccessful in making the ball segway transfer while balancing. By contrast, the double-loop control scheme was successful for point-to-point transferring of the ball segway while balancing.

To demonstrate the ability of the ball segway to move in any direction on the floor, it was commanded to move from point to point and track a desired straight line. This test validated the capabilities of the control system.

The first-position control experiment is conducted to determine the transferring performance of the robot on the floor. The state-weighting matrix $\mathbf{Q}$ and factor R are redefined in this experiment. By varying the matrix $\mathbf{Q}$ and factor

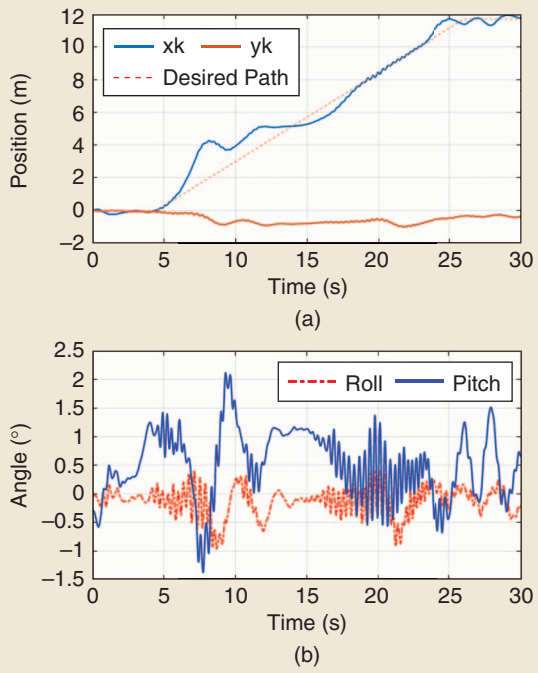


FIGURE 15 Experimental results of transferring control along the x-axis. The ball segway is commanded to move from its origin along the x-axis to the goal position ($x = 11.5$ m). (a) The position of the ball on the floor while transferring. (b) Tilt angles of the body while transferring.

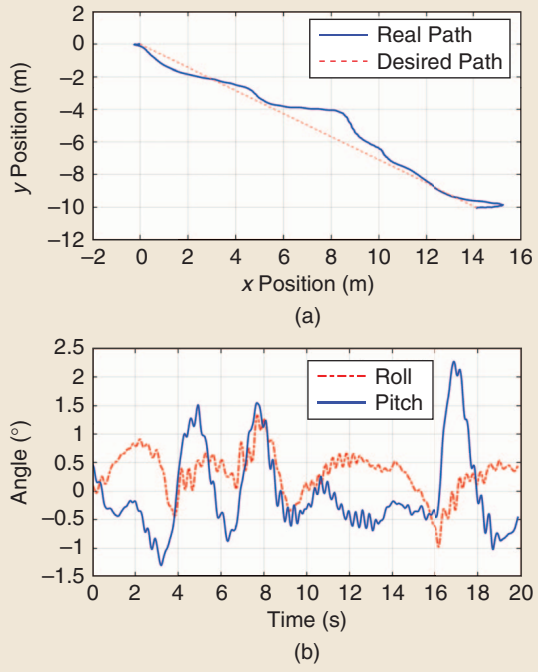


FIGURE 16 Experimental results of straight-line tracking control. The straight-line path is generated by commanding the ball segway to move at a constant velocity for a period of 20 s. (a) The trajectory of the ball for tracking the straight line. (b) The tilt angles of the body during straight-line tracking.

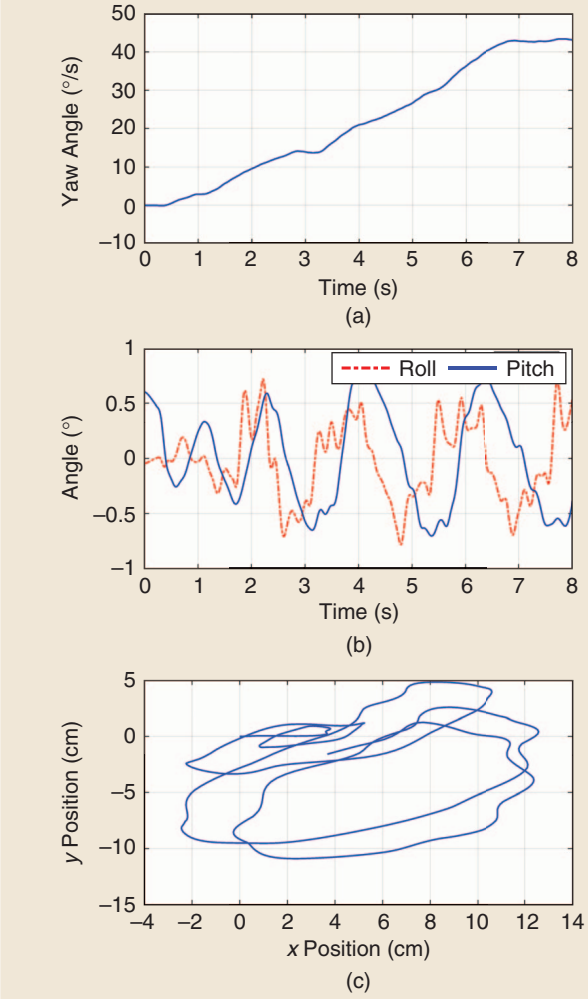


FIGURE 17 Experimental results of rotating control around the vertical axis. Based on the two-dimensional model for the x–y plane, a proportional-derivative controller is utilized to control the rotation of the ball segway around the z-axis. (a) The yaw angle of the body. (b) Tilt angles of the body. (c) The position of the ball on the floor.

R, the feedback control gains changed. The ball segway is required to transfer along the x-axis from the initial position $(x_k, y_k) = (0, 0)$ to the target position $(x_k, y_k) = (11.5\text{ m}, 0\text{ m})$. The desired position is ramped to obtain a sufficient pitch angle of the body. The experimental results are shown in Figure 15. The position error is nearly 15 cm in the x-direction, as shown in Figure 15(a). The roll and pitch angle trajectories of the body during the point-to-point transfer are shown in Figure 15(b).

The second-position control experiment aims to transfer the robot from one direction and follow a desired straight line. Figure 16(a) shows the desired trajectory and the position response of the ball segway. In Figure 16(a), the robot is commanded to move and track the desired path from the origin to the goal point $(x_k, y_k) = (14\text{ m}, -10\text{ m})$ within 20 s. The body tilt angles are shown in Figure 16(b). A 20-cm position error occurs at the final position. Unknown

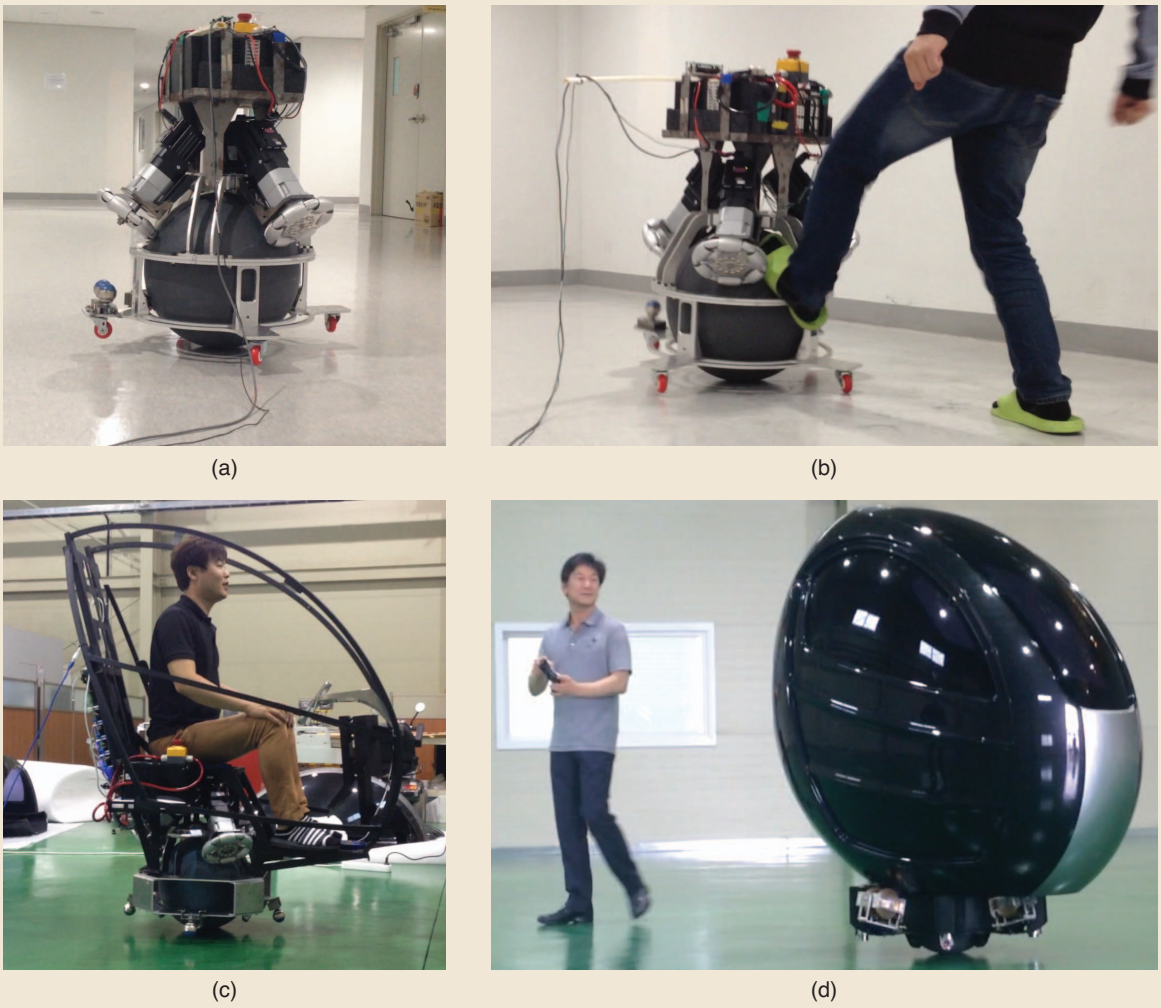


FIGURE 18 Photos from a motion experiment. (a) Balancing the ball segway on the floor. (b) The robustness of the double-loop control system against a kick. (c) The ball segway transporting a person on the floor. (d) Controlling movement and rotation using a joystick.

disturbances, system parameters and modeling uncertainties, floor conditions, and undefined friction cause control disorder and make the system unstable.

Rotating Control

An experiment is conducted to investigate the rotating motion of the ball segway. The yaw angle and angular velocity are directly measured using the IMU sensor. A PD controller is deployed to control the rotation of the ball segway around the z-axis. The rotational rate of the ball segway can be changed by varying the control gains of the PD controller. Figure 17(a) shows the yaw angle while the ball segway balances on the floor, as shown in Figure 17(b). Figure 17(c) shows the position of the ball in the small moving distance on the floor while rotating around the vertical axis. Rotation experiment results are available in [27].

Figure 18 shows the photos from the ball segway operation experiment. The LQR and PI double-loop controller shown in the aforementioned results indicates satisfactory control performance. Thus, this controller is suitable for real-time control of the ball segway.

CONCLUSIONS

The structure of a ball segway is presented in this study, where the mechanism and hardware architecture of the ball segway are illustrated. The nonlinear dynamic model of the 2-D ball segway is obtained using the Euler–Lagrange equation and the decoupling method. The nonlinear motion equations of the robot are then linearized around an equilibrium point. Based on the double-loop control system and the linear model of the ball segway, the balancing, robustness, and transferring controllers have been synthesized by the double-loop control system that consists of a PI plus feedforward inner-loop controller and an LQR outer-loop controller. Simulations and experiments are conducted, and their results show that the double-loop control system successfully controls its balance in a real environment. The experimental results indicate that the ball segway can simultaneously track the desired trajectory and maintain its

balancing stability. Moreover, the robust properties of the controller under large external disturbances are inherited from the double-loop control scheme. The experimental results demonstrate that the ball segway not only simultaneously tracks the desired trajectory and maintains balancing but also rotates around the vertical axis.

ACKNOWLEDGMENTS

This research was supported by the Development of Robot System for Inspecting and Cleaning an Industrial Electrostatic Precipitators Program through the MKE (Grant Number: 10067781) and by the Development of Autonomous Navigation Algorithm for a High-Speed Unmanned Surface Vehicle Program of LIG Nex1 Co. Ltd. It was also supported by the Senior-Friendly Product R&D Program funded by the Ministry of Health and Welfare through the Korea Health Industry Development Institute (HI15C1027).

AUTHOR INFORMATION

Dinh Ba Pham received the B.E. and M.S. degrees in mechanical engineering from Vietnam Maritime University, Hai Phong, Vietnam, in 2007 and 2012, respectively. He is currently a Ph.D. student in mechanical engineering at Kyung Hee University, Yongin, Korea. His research interests include robotics, dynamics, applied nonlinear control, and control of industrial machines.

Hyung Kim received the B.E. degree in mechanical engineering from NILE, Korea, and the M.S. degree in mechanical engineering from Kyung Hee University, Korea, in 2008 and 2010, respectively. He is currently a Ph.D. student in mechanical engineering at Kyung Hee University. His research interests include applied mobile robots, environment recognition, and intelligent control.

Jaejun Kim received the B.E degree from Dong-Eui University, Korea, and M.S. degree in mechanical engineering from Kyung Hee University, Korea, in 2007 and 2009, respectively, where he is currently a Ph.D. student. His research interests include robotics and control of industrial machines.

Soon-Geul Lee (sglee@khu.ac.kr) received the B.E. degree in mechanical engineering from Seoul National University, Korea, in 1983, the M.S. degree in production engineering from KAIST, Seoul, Korea, in 1985, and the Ph.D. in mechanical engineering from the University of Michigan in 1993. Since 1996, he has been a professor in the Department of Mechanical Engineering of Kyung Hee University, Yongin, Korea. His research interests include robotics and automation, mechatronics, intelligent control, and biomechanics.

REFERENCES

[1] T. B. Lauwers, G. A. Kantor, and R. L. Hollis, "A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2006, pp. 2884–2889.

[2] U. Nagarajan, G. Kantor, and R. Hollis, "The ballbot: An omnidirectional balancing mobile robot," *Int. J. Robotics Res.*, vol. 33, no. 6, pp. 917–930, 2014.

[3] M. Kumaga and T. Ochiai, "Development of a robot balanced on a ball & application of passive motion to transport," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2009, pp. 4106–4111.

[4] L. Hertig, D. Schindler, M. Bloesch, C. D. Remy, and R. Siegwart, "Unified state estimation for a ballbot," in *Proc. IEEE Int. Conf., Robotics and Automation,* 2013, pp. 2471–2476.

[5] M. Akbari, H. Zabihi Kheibari, and A. S. Moosavi Nejad, "Timing belt gearbox in Ballbot robot," in *Proc. 1st RSI/ISM Int. Conf. Robotics and Mechatronics,* 2013, pp. 431–436.

[6] N. Aphiratsakun, P. K. Remi Nordeng, M. Suikkanen, and N. Lorpatanakasem, "Implementation of AU balancing ballbot," in *Proc. Int. Electrical Engineering Congr.,* 2014, pp. 1–4.

[7] T. Endo and Y. Nakamura, "An omnidirectional vehicle on a basketball," in *Proc. 12th Int. Conf. Advanced Robotics*, 2005, pp. 573–578.

[8] H. Y. Han, T. Y. Han, and H. S. Jo, "Development of omnidirectional self-balancing robot," in *Proc. IEEE Int. Symp. Robotics and Manufacturing Automation*, 2014, pp. 57–62.

[9] B. Vaidya, M. Shomin, R. Hollis, and G. Kantor, "Operation of the ballbot on slopes and with center-of-mass offsets," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2015, pp. 2383–2388.

[10] C. C. Tsai, C. K. Chan, and L. C. Kuo, "LQR motion control of a ball-riding robot," in *Proc. IEEE/ASME Int. Conf. Advanced Intelligent Mechatronics*, 2012, pp. 861–866.

[11] T. Ching-Chih, J. Ming-Han, C. Cheng-Kai, L. Ching-Wen, and C. Siang-Jyun, "Self-balancing and position control using multi-loop approach for ball robots," in *Proc. Int. Conf. System Science and Engineering*, 2010, pp. 251–256.

[12] K. Sukvichai and M. Parnichkun, "Double-level ball-riding robot balancing: From system design, modeling, controller synthesis, to performance evaluation," *Mechatronics*, vol. 24, pp. 519–532, July 2014.

[13] L. Ching-Wen, T. Ching-Chih, L. Yi Yu, and C. Cheng-Kai, "Dynamic modeling and sliding-mode control of a Ball robot with inverse mouse-ball drive," in *Proc. Annu. Conf. Society Instrument and Control Engineers Japan*, 2008, pp. 2951–2955.

[14] R. A. Garcia-Garcia and M. Arias-Montiel, "Linear controllers for the NXT ballbot with parameter variations using linear matrix inequalities," *IEEE Control Syst.*, vol. 36, no. 3, pp. 121–136, June 2016.

[15] C. Chih-Hui and T. Wen-Ru, "Design and implementation of an omnidirectional spherical mobile platform," *IEEE Trans. Ind. Electron.*, vol. 62, pp. 1619–1628, Mar. 2015.

[16] J. Jian, M. Wang, N. Lv, Y. Yang, K. Zhang, Y. Li, "Standing-up control and ramp-climbing control of a spherical wheeled robot," in *Proc. 13th Int. Conf. Control Automation Robotics and Vision*, 2014, pp. 1386–1391.

[17] C. H. Chiu and Y. F. Peng, "Omni-directional spherical mobile system control," in *Proc. 9th Asian Control Conf.*, 2013, pp. 1–4.

[18] C.-H. Chiu, Y.-F. Peng, W.-R. Tsai, and M.-H. Chou "Design of an omnidirectional spherical robot: Using fuzzy control," in *Proc. Int. Multi-Conf. Engineers and Computer Scientists*, 2009, vol. I, p. 6.

[19] T. Ching-Chih, C. Chang Hsuan, T. Feng-Chun, and H. Kao-Shing, "Adaptive RFWCMAC cooperative formation control for multiple ballbots incorporated with coupling dynamics," in *Proc. Int. Conf. Informative and Cybernetics Computational Social Systems*, 2015, pp. 59–65.

[20] S.-E. Choi, Y.-J. Kim, and J.-J. Ko. (2015). Pan and history of the tech-campaign of Hankook Tire Co. [Online]. Available: https:// www.youtube .com/watch?v=iGwidDckzik

[21] P. D. Ba, S.-G. Lee, S. Pack, J. Kim, and M.-K. Lee, "Balancing and translation control of a ball Segway that a human can ride," in *Proc. 16th Int. Conf. Control Automation and Systems,* 2016, pp. 477–480.

[22] A. Weiss, R. G. Langlois, and M. J. D. Hayes, "The effects of dual row omnidirectional wheels on the kinematics of the Atlas spherical motion platform," *Mechanism Mach. Theory*, vol. 44, pp. 349–358, May 2008.

[23] K. v. d. Blonk. (2014, July). Modeling and control of a ball-balancing robot. M.S. thesis, Internship and Master Elect. Eng. Math. Computer Sci., Univ. Twente, Alten. [Online]. Available: https://www.utwente.nl/en/eemcs/sacs/teaching/Thesis/blonk.pdf

[24] H. E. A. Ibrahim, F. N. Hassan, and A. O. Shomer, "Optimal PID control of a brushless DC motor using PSO and BF techniques," *Ain Shams Eng. J.*, vol. 5, pp. 391–398, Nov. 2013.

[25] K. Ogata, *Modern Control Engineering*, 5th ed. New York: Pearson, 2010.

[26] D. B. Pham and S.-G. Lee. (2017). Ball segway balancing and robust control against external disturbance. [Online]. Available: https://www .youtube.com/watch?v=1Xp0yDVX1pw

[27] D. B. Pham and S.-G. Lee. (2017). The ball segway test for balancing, rotating, and moving. [Online]. Available: https://www. youtube.com/watch?v=yyuXKmpDweQ