

Article

Neural Network Control for Trajectory Tracking and Balancing of a Ball-Balancing Robot with Uncertainty

Hyo-Geon Jang ¹ , Chang-Ho Hyun ² and Bong-Seok Park ^{1,2,*} ¹ Department of Future Convergence Engineering, Kongju National University, Cheonan 31080, Korea; hyosky@smail.kongju.ac.kr² Department of Electrical, Electronic, and Control Engineering, Kongju National University, Cheonan 31080, Korea; hyunch@kongju.ac.kr

* Correspondence: bspark@kongju.ac.kr; Tel.: +82-41-521-9164

Abstract: In this paper, a neural-network-based control method to achieve trajectory tracking and balancing of a ball-balancing robot with uncertainty is presented. Because the ball-balancing robot is an underactuated system and has nonlinear couplings in the dynamic model, it is challenging to design a controller for trajectory tracking and balancing. Thus, various approaches have been proposed to solve these problems. However, there are still problems such as the complex control system and instability. Therefore, the objective of this paper was to propose a solution to these problems. To this end, we developed a virtual angle-based control scheme. Because the virtual angle was used as the reference angle to achieve trajectory tracking while keeping the balance of the ball-balancing robot, we could solve the underactuation problem using a single-loop controller. The radial basis function networks (RBFNs) were employed to compensate uncertainties, and the controller was designed using the dynamic surface control (DSC) method. From the Lyapunov stability theory, it was proven that all errors of the closed-loop control system were uniformly ultimately bounded. Therefore, the control system structure was simple and ensured stability in achieving simultaneous trajectory tracking and balancing of the ball-balancing robot with uncertainty. Finally, the simulation results are given to verify the performance of the proposed controller through comparison results. As a result, the proposed method showed a 19.2% improved tracking error rate compared to the existing method.

Keywords: ball-balancing robot; underactuated system; virtual angle; dynamic surface control; neural network



Citation: Jang, H.-G.; Hyun, C.-H.; Park, B.S. Neural Network Control for Trajectory Tracking and Balancing of a Ball-Balancing Robot with Uncertainty. *Appl. Sci.* **2021**, *11*, 4739. <https://doi.org/10.3390/app11114739>

Academic Editor: Alessandro Gasparetto

Received: 7 May 2021

Accepted: 19 May 2021

Published: 21 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Mobile robots have been attracting attention in many fields due to their various applications. Therefore, various techniques have been proposed to operate mobile robots, such as automatic control and a brain–computer interface (BCI) that connects the brain and a computer to control the robot [1]. In this paper, we focused on the automatic control of a ball-balancing robot, which is a type of mobile robot. The ball-balancing robot is an omnidirectional mobile robot, the body of which is placed over a spherical wheel. Because of this characteristic, a control strategy different from traditional methods for mobile robots was required to achieve simultaneous trajectory tracking and balancing. However, designing the controller was challenging due to several problems such as underactuation and nonlinear coupling.

Early studies used single-loop control methods such as proportional–integral–derivative (PID) [2] and a linear quadratic regulator (LQR) [3], which was only possible for balancing the robot. To enable both trajectory tracking and balancing of the robot, multi-loop control methods such as PI-LQR [4] and PD-PID [5] were presented using independent inner- and outer-loops. However, the above papers were based on the linearized model, which requires a narrow operating range. Although the partial feedback linearization (PFL)

technique [6] was proposed to relax this restriction, the control system using the multi-loop method becomes complex due to additional feed-forward controllers for robustness against uncertainties [7].

Sliding mode control (SMC) is a representative nonlinear control method that is robust to uncertainty. Thus, the hierarchical sliding mode control (HSMC) method [8], which is composed of combinations of independent sliding surfaces, was proposed to address underactuated systems. The HSMC method based on the two-dimensional model of the ball-balancing robot was presented in [9] and then applied to the aggregated model to consider the combined effects [10]. An integral backstepping technique using the HSMC structure was proposed in [11], and an artificial neural network-based method [12] was presented to deal with uncertainty and input saturation. The hierarchical structure used in the above papers can solve the underactuation problem and has robustness for uncertainties. However, the hierarchical structure has the local minimum problem [13], which may not guarantee the convergence of all errors.

Therefore, a different approach was needed to achieve simultaneous trajectory tracking and balancing for the ball-balancing robot. In this regard, we proposed a virtual angle-based approach [14]. The virtual angle composed of the tracking errors was used as a reference angle to achieve simultaneous trajectory tracking and balancing using the single-loop control system. However, the upper bounds of uncertainties for robustness should be known, which may be impossible in the real world. Artificial intelligence can be a solution to address this problem. In [15], deterministic artificial intelligence (DAI) was used for underactuated systems with nonlinear couplings. In [16], a artificial neural network was employed to deal with model uncertainty and the external disturbance of underactuated surface vessels. The field of non-stochastic or DAI involves the assertion of self-awareness statements and uses optimal learning to compensate the deleterious effects of error sources [15]. On the other hand, an artificial neural network (ANN), which is in the field of stochastic artificial intelligence, does not need to devise an algorithm to perform a particular task because the learning process of a neural network takes place by itself [17].

Motivated by these observations, we proposed a neural network-based control for trajectory tracking and balancing of the ball-balancing robot with uncertainty. A virtual angle-based control method was used to design a single-loop control system to achieve trajectory tracking and balancing of the robot. Unlike in [14], the radial basis function networks (RBFNs) [18], a kind of ANN, were used to compensate the uncertainties without knowing their upper bounds, and the dynamic surface control (DSC) method [19] was employed to design the controller. From the Lyapunov stability theory, it was proven that all errors of the closed-loop control system were uniformly ultimately bounded. Finally, the performance of the proposed method was demonstrated through the simulation results.

The main contributions of this paper are summarized as follows:

1. The trajectory tracking and balancing of the ball-balancing robot was achieved using a single-loop controller;
2. It was possible to implement a control system without any knowledge of the upper bounds of uncertainties.

This paper is organized as follows. In Section 2, the dynamics of the ball-balancing robot and the RBFNs are described, and the neural network-based controller is developed for trajectory tracking and balancing of the ball-balancing robot. In Section 3, the simulation results are presented to demonstrate the performance of the proposed method, and the discussion is given in Section 4.

2. Materials and Methods

In this section, we describe the system model of the ball-balancing robot for formulating the problems and introduce the RBFNs for compensating the uncertainty.

2.1. Dynamics of the Ball-Balancing Robot

Consider the ball-balancing robot shown in Figure 1. Assuming that there is no slip between the sphere and the floor and between the driving wheels and the sphere, the dynamic equation of the ball-balancing robot can be expressed as [12]:

$$M(q)\ddot{q} + C(\dot{q}, q)\dot{q} + f(\dot{q}) + g(q) = B(q)u. \quad (1)$$

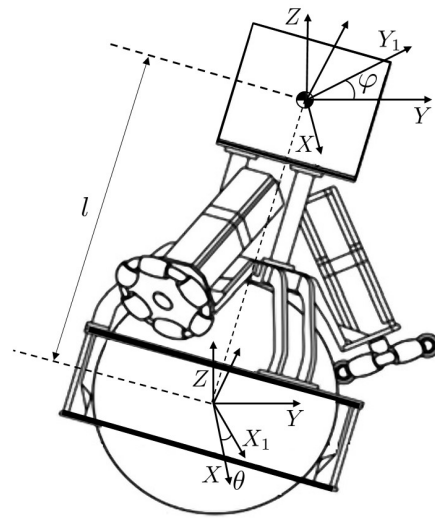


Figure 1. Spatial model of the ball-balancing robot [10].

Here, $M(q) \in \mathbb{R}^{4 \times 4}$ is the symmetric inertia matrix, $C(\dot{q}, q) \in \mathbb{R}^{4 \times 4}$ represents the Coriolis and centripetal matrix, $f(\dot{q}) \in \mathbb{R}^4$ is the friction matrix, $g(q) \in \mathbb{R}^4$ is the gravity vector, $B(q) \in \mathbb{R}^{4 \times 2}$ denotes control input matrix, and $u \in \mathbb{R}^2$ represents the control input vector. These terms are given by:

$$M(q) = \begin{bmatrix} m_1 & m_2 & m_3 & m_4 \\ m_2 & m_5 & m_6 & m_7 \\ m_3 & m_6 & m_8 & 0 \\ m_4 & m_7 & 0 & m_9 \end{bmatrix},$$

$$C(\dot{q}, q) = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & 0 & c_{44} \end{bmatrix},$$

$$f(\dot{q}) = [\zeta_x \dot{x}, \zeta_y \dot{y}, \zeta_\theta \dot{\theta}, \zeta_\phi \dot{\phi}]^\top,$$

$$g(q) = [0, 0, g_3, g_4]^\top, \quad u = [u_x, u_y]^\top,$$

$$B(q) = [B_s^\top, B_b^\top]^\top, \quad q = [x, y, \theta, \phi]^\top,$$

$$B_s = \frac{1}{r_w} \begin{bmatrix} 0 & -\cos \phi \\ \cos \theta & \sin \phi \cos \theta \end{bmatrix}^\top, \quad B_b = \frac{r_s}{r_w} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^\top,$$

where the X - and Y -axis positions of the ball are denoted by x and y , respectively, pitch and roll are denoted by θ and ϕ , respectively, and u_x and u_y denote the control inputs of the virtual acting wheels by the three omnidirectional wheels, respectively. The terms in ζ_i ($i = x, y, \theta, \phi$) are viscous coefficients. r_s and r_w denote the ball radius and omnidirectional wheel radius, respectively. It is noted that control input matrices B_s and B_b are invertible with $|\theta|, |\phi| < \pi/2$. The terms in $M(q)$, $C(\dot{q}, q)$, and $g(q)$ are given in Appendix A.

The dynamic equation in (1) can be decoupled into two subsystems of the position and the angle as follows:

$$\ddot{\mathbf{q}}_s = \mathbf{F}_s + \Delta\mathbf{F}_s + \mathbf{G}_s\mathbf{u}, \quad (2)$$

$$\ddot{\mathbf{q}}_b = \mathbf{F}_b + \Delta\mathbf{F}_b + \mathbf{G}_b\mathbf{u}. \quad (3)$$

In these expressions, $\mathbf{q}_s = [x, y]^\top$, $\mathbf{q}_b = [\theta, \varphi]^\top$, and the matrices $\mathbf{F}_j$, $\Delta\mathbf{F}_j$ and $\mathbf{G}_j$ ($j = s, b$) are given in Appendix B. $\Delta\mathbf{F}_j$ represents the model uncertainties including viscous damping coefficients ζ_i ($i = x, y, \theta, \varphi$).

The control objective was to design the control input $\mathbf{u}$ satisfying that $\lim_{t \rightarrow \infty} \|\mathbf{q}_s - \mathbf{q}_{sr}\| = \mu_0$ and $\lim_{t \rightarrow \infty} \|\mathbf{q}_b - \mathbf{q}_{br}\| = \mu_1$, where $\mathbf{q}_{sr} = [x_r, y_r]^\top$ and $\mathbf{q}_{br} = [0, 0]^\top$ are the reference vectors for trajectory tracking and balancing, respectively, and μ_0 and μ_1 are arbitrary positive constants.

2.2. Radial Basis Function Networks

We employed the RBFNs [20] to approximate $\Delta\mathbf{F}_b$, and it can be expressed as:

$$\begin{aligned} \Delta\mathbf{F}_b &= [\Delta F_{b1}, \Delta F_{b2}]^\top \\ &= [\Phi_1(\chi_1)\omega_1^* + \varepsilon_1, \Phi_2(\chi_2)\omega_2^* + \varepsilon_2]^\top, \end{aligned} \quad (4)$$

where $\chi_i \in \mathbb{R}^n$ ($i = 1, 2$) is the input vector of the RBFN, $\omega_i^* \in \mathbb{R}^m$ is the optimal weight vector, m is the node number, ε_i denotes the reconstruction error, and $\Phi_i(\chi_i) = [\Phi_{i1}(\chi_i), \dots, \Phi_{im}(\chi_i)]$ with:

$$\Phi_{ij}(\chi_i) = \exp\left(\frac{-\|\chi_i - \mu_{ij}\|^2}{\eta_{ij}^2}\right), \quad j = 1, 2, \dots, m. \quad (5)$$

Here, $\mu_{ij} \in \mathbb{R}^n$ is the center of the receptive field and η_{ij} is the width of the Gaussian function.

Assumption 1. There exists an ideal constant weight vector ω_i^* such that $|\varepsilon_i| \leq \varepsilon_M$ with a positive constant ε_M for $i = 1, 2$, and the optimal weight vector is defined as the value of $\hat{\omega}_i$ that minimizes $|\varepsilon_i|$, i.e.,

$$\omega_i^* = \arg \min_{\hat{\omega}_i \in \mathbb{R}^m} \left\{ \sup_{\chi_i \in \mathbb{R}^n} |\Delta F_{bi} - \Phi_i(\chi_i)\hat{\omega}_i| \right\}. \quad (6)$$

Assumption 2. The optimal weight vector is bounded such that $\|\omega_i^*\| \leq \omega_M$ with a positive constant ω_M .

2.3. Controller Design

The ball-balancing robot model given in (2) and (3) is underactuated and has uncertainties. To deal with these problems, we introduced a virtual angle that can achieve both the trajectory tracking and balancing of the robot and designed a neural-network-based controller using the DSC method step-by-step.

Let us define the following errors:

$$\mathbf{e}_s = [e_x, e_y]^\top = \mathbf{q}_s - \mathbf{q}_{sr}, \quad (7)$$

$$z_1 = \tanh \mathbf{q}_b - \boldsymbol{\theta}_f,$$

$$\mathbf{z}_2 = [z_{21}, z_{22}]^\top = \dot{\mathbf{q}}_b - \boldsymbol{\alpha}_f, \quad (8)$$

where θ_f and α_f are the signals filtered by $\tau_1^2 \ddot{\theta}_f + 2\tau_1 \dot{\theta}_f + \theta_f = \theta_v$ and $\tau_2 \ddot{\alpha}_f + \alpha_f = \alpha$ with positive design parameters τ_1 and τ_2 . In these expressions, α denotes the virtual control input and θ_v denotes the virtual angle for the underactuation problem given by:

$$\theta_v = \lambda_\theta s, \quad s = \dot{e}_s + \lambda e_s, \quad (9)$$

where λ_θ is a diagonal matrix and $\lambda > 0$ is a design parameter.

Remark 1. In (8), the boundedness of z_1 implies that θ_f and θ_v are bounded. This leads to the boundedness of the tracking error e_s from (9). If the tracking error converges to zero, it is easily shown that the body angle q_b of the robot also converges to zero from (8) and (9). Therefore, we can solve the underactuation problem.

Remark 2. The first and second derivatives of the virtual angle θ_v are required to design a controller. By the definition of the virtual angle in (9), it includes the control input u , which is impossible to implement the controller. To avoid this problem, we used the virtual angle θ_f passed through the second-order filter.

Differentiating both sides of (8) using (3) gives the following error dynamics:

$$\dot{z}_1 = H_1(z_2 + \alpha_f) - \dot{\theta}_f, \quad (10)$$

$$\dot{z}_2 = F_b + \Delta F_b + G_b u - \dot{\alpha}_f, \quad (11)$$

where $H_1 = \text{diag}(\text{sech}^2\theta, \text{sech}^2\varphi)$ and $\text{diag}(\cdot)$ denotes a diagonal matrix. Note that H_1 is invertible.

Step 1: Consider the Lyapunov function candidate as:

$$V_1 = \frac{1}{2} z_1^\top z_1. \quad (12)$$

Let ω be the filter error of the virtual control input, i.e., $\omega = \alpha - \alpha_f$. Differentiating both sides of (12) along (10) yields:

$$\dot{V}_1 = z_1^\top (H_1(z_2 + \alpha - \omega) - \dot{\theta}_f). \quad (13)$$

From (13), we chose the virtual control input vector α to be:

$$\alpha = H_1^{-1}(\dot{\theta}_f - K_1 z_1), \quad (14)$$

where K_1 is a diagonal positive definite matrix.

Step 2: Consider the Lyapunov function candidate as:

$$V_2 = \frac{1}{2} (z_2^\top z_2 + \omega^\top \omega). \quad (15)$$

The time derivative of (15) along (11) is:

$$\dot{V}_2 = z_2^\top (F_b + \Delta F_b + G_b u - \dot{\alpha}_f) + \omega^\top (\dot{\alpha} - \tau_2^{-1} \omega). \quad (16)$$

From (16), we chose the actual control vector u to be:

$$u = G_b^{-1} \left(-F_b - [\Phi_1(\chi_1) \hat{\omega}_1, \Phi_2(\chi_2) \hat{\omega}_2]^\top + \dot{\alpha}_f - K_2 z_2 - H_1^\top z_1 \right), \quad (17)$$

where K_2 is a diagonal positive definite matrix and $\hat{\omega}_i (i = 1, 2)$ denotes the estimate of the optimal weight vector ω_i^* for the RBFNs.

Remark 3. Because the ball-balancing robot is operated by the three driving motors acting on the spherical wheel, it needs torque conversion between the control input $\mathbf{u}$ and the actual motors for the implementation (see the details in [7]).

Step 3: Consider the total Lyapunov function candidate as:

$$V_T = V_1 + V_2 + \frac{1}{2} \sum_{i=1}^2 \tilde{\omega}_i^\top \Gamma_i^{-1} \tilde{\omega}_i, \quad (18)$$

where $\tilde{\omega}_i = \omega_i^* - \hat{\omega}_i$ and Γ_i is a diagonal positive definite matrix.

The time derivative of (18) along (13), (14), (16), and (17) is:

$$\begin{aligned} \dot{V}_T = & -z_1^\top K_1 z_1 - z_2^\top K_2 z_2 + z_2^\top \begin{bmatrix} \Phi_1(\chi_1) \tilde{\omega}_1 + \varepsilon_1 \\ \Phi_2(\chi_2) \tilde{\omega}_2 + \varepsilon_2 \end{bmatrix} \\ & - \tau_2^{-1} \omega^\top \omega - z_1^\top H_1 \omega + \omega^\top \dot{\alpha} - \sum_{i=1}^2 \tilde{\omega}_i^\top \Gamma_i^{-1} \dot{\tilde{\omega}}_i. \end{aligned} \quad (19)$$

From (19), we chose the update law $\dot{\hat{\omega}}_i$ to be:

$$\dot{\hat{\omega}}_i = \Gamma_i (\Phi_i(\chi_i)^\top z_{2i} - \sigma_i \hat{\omega}_i), \quad i = 1, 2, \quad (20)$$

where σ_i is a positive design parameter.

Substituting (20) into (19) yields:

$$\dot{V}_T = - \sum_{i=1}^2 \left[z_i^\top K_i z_i + \sigma_i \tilde{\omega}_i^\top (\tilde{\omega}_i - \omega_i^*) \right] - \tau_2^{-1} \omega^\top \omega - z_1^\top H_1 \omega - z_2^\top \varepsilon + \omega^\top \dot{\alpha}, \quad (21)$$

where $\varepsilon = [\varepsilon_1, \varepsilon_2]^\top$.

3. Results

In this section, the stability of the control system designed in Section 2 is analyzed using the Lyapunov stability theory, and the main result of this paper is given. Moreover, the simulation results are provided to verify the performance of the proposed method.

3.1. Stability Analysis

The main result of this paper is given by the following theorem.

Theorem 1. Consider the ball-balancing robot model given in (2) and (3). Under Assumptions 1 and 2, if the control inputs α and $\mathbf{u}$ given in (14) and (17) with the update law (20) are applied to the ball balancing model, then for any initial conditions satisfying $V_T(0) \leq p$, where p is a positive constant, it is ensured that all error signals in the closed-loop system are uniformly ultimately bounded and can be made arbitrarily small.

Proof. From (14), one can show that $\|\dot{\alpha}\| \leq \eta(z_1, z_2, \omega, \theta_f, \dot{\theta}_f, \ddot{\theta}_f)$ for some continuous functions η . Consider a set $Q := \{(\theta_f, \dot{\theta}_f, \ddot{\theta}_f) : \|\theta_f\| + \|\dot{\theta}_f\| + \|\ddot{\theta}_f\| \leq K_0\}$ and a set $A := \{(z_1, z_2, \omega, \tilde{\omega}_1, \tilde{\omega}_2) : z_1^\top z_1 + z_2^\top z_2 + \omega^\top \omega + \tilde{\omega}_1^\top \Gamma_1^{-1} \tilde{\omega}_1 + \tilde{\omega}_2^\top \Gamma_2^{-1} \tilde{\omega}_2 \leq 2p\}$ where K_0 is a positive constant. Since Q and A are compact sets, there exists a positive constant η_M such that $\eta(\cdot) \leq \eta_M$ on $A \times Q$. Then, by Assumptions 1 and 2 and Young's inequality, (21) can be rewritten as:

$$\begin{aligned} \dot{V}_T \leq & - \sum_{i=1}^2 \left[z_i^\top \left(K_i - \frac{1}{2} I_2 \right) z_i + \frac{1}{2} \sigma_i \tilde{\omega}_i^\top \tilde{\omega}_i \right] \\ & - \omega^\top \left(\tau_2^{-1} - \frac{1}{2} \right) I_2 \omega + \varepsilon_M^2 + \frac{1}{2} \sigma_1 \omega_M^2 + \frac{1}{2} \sigma_2 \omega_M^2 + \frac{1}{2} \eta_M^2, \end{aligned} \quad (22)$$

where $I_2 \in \mathbb{R}^{2 \times 2}$ denotes the identity matrix.

Choosing $K_1 = \frac{1}{2}I_2 + K_1^*$, $K_2 = \frac{1}{2}I_2 + K_2^*$, and $\tau_2^{-1} = \frac{1}{2} + \tau_2^*$ with K_1^* , K_2^* , $\tau_2^* > 0$ results in:

$$\dot{V}_T \leq -c_0 V_T + c_1, \quad (23)$$

where $c_0 = \min\{\lambda_{K_1^*}, \lambda_{K_2^*}, \tau_2^*, \sigma_1 \lambda_{\Gamma_1^{-1}}, \sigma_2 \lambda_{\Gamma_2^{-1}}\}$, $c_1 = \varepsilon_M^2 + \frac{1}{2}\sigma_1 \omega_M^2 + \frac{1}{2}\sigma_2 \omega_M^2 + \frac{1}{2}\eta_M^2$, and $\lambda_{(\cdot)}$ denotes the minimum eigenvalue of the matrix $(\cdot)$.

Integrating both sides of (23) yields:

$$V(t)_T \leq \left(V_T(0) - \frac{c_1}{c_0}\right)e^{-c_0 t} + \frac{c_1}{c_0}, \quad (24)$$

From (24), the boundedness of z_1 is guaranteed, and it leads to the boundedness of the virtual angle θ_v .

Integrating both sides of (9) yields:

$$\|e_s(t)\| \leq \min\{\|e_s(0)\|, \|\lambda^{-1} \lambda_\theta^{-1} \theta_v\|\}. \quad (25)$$

Therefore, all errors in the closed-loop system are uniformly ultimately bounded and can be made arbitrarily small by properly choosing the design parameters. This completes the proof. $\square$

3.2. Simulation Results

This section presents the simulation results on the proposed method. The model parameters are given in Table 1. The design parameters were chosen as $K_1 = \text{diag}(10, 10)$, $K_2 = \text{diag}(10, 10)$, $\lambda = 2$, $\lambda_\theta = \text{diag}(-0.1, 0.1)$, $\sigma_1 = \sigma_2 = 0.1$, and $\tau_1 = \tau_2 = 0.1$. The initial states of the robot were taken as $q = [-0.2, 0.2, 0, 0]^T$. The friction parameters were chosen as $\zeta_x = \zeta_y = 2 \text{ N} \cdot \text{s/m}$ and $\zeta_\theta = \zeta_\varphi = 1 \text{ N} \cdot \text{s/rad}$. To consider perturbed dynamics, we assumed that we knew the nominal parameters as follows: $m_b = 116 \text{ kg}$, $l = 0.23 \text{ m}$, $I_{bx} = 16.25 \text{ kg} \cdot \text{m}^2$, and $I_{by} = 15.85 \text{ kg} \cdot \text{m}^2$. To verify the performance of the proposed method, the reference trajectory was composed of straight lines and curves, and the proposed controller (PC) was compared with the integral backstepping hierarchical sliding mode control (IBHSMC) method [12].

Table 1. Parameters of the ball-balancing robot [12].

Symbol	Parameter	Value
m_b	Body mass	116 kg
l	Body length	0.23 kg
I_{bx}	Moment of inertia of the body about the roll axis	16.25 kg · m ²
I_{by}	Moment of inertia of the body about the pitch axis	15.85 kg · m ²
r_w	Omnidirectional wheel radius	0.1 kg
m_w	Omnidirectional wheel mass	0.56 kg
I_w	Moment of inertia of the omnidirectional wheel	0.26 kg
m_s	Ball mass	11.4 kg
r_s	Ball radius	0.19 kg
I_s	Moment of inertia of the ball	0.165 kg
α	Zenith angle	56°

The simulation results on the trajectory tracking are shown in Figure 2. It can be seen that both methods had precise tracking performance for straight paths and slight errors for curved paths. Figure 3 shows the detailed simulation results. The trajectory

tracking errors of the ball-balancing robot are shown in Figure 3a,b, and the body angles of the robot are depicted in Figure 3c,d. It can be seen that neither method made a significant difference in balancing performance. On the other hand, in trajectory tracking, the proposed method showed a 19.2% improved error rate compared to the IBHSMC method. Figure 4a,b shows the control inputs and estimates of the optimal weights for the RBFNs, respectively. Table 2 shows the root mean square (RMS) of the tracking errors for various control gains. This result sufficiently validated that the tracking and balancing errors can be made arbitrarily small by properly choosing the design parameters. From these results, we could demonstrate that the proposed method was effective in achieving simultaneous trajectory tracking and balancing of the ball-balancing robot.

Table 2. RMS of tracking errors according to the control gain.

Gain	Nominal	Perturbed
$K_1 = \text{diag}(0.1, 0.1)$	0.1299	0.2108
$K_1 = \text{diag}(1, 1)$	0.1278	0.1355
$K_1 = \text{diag}(10, 10)$	0.1271	0.1244

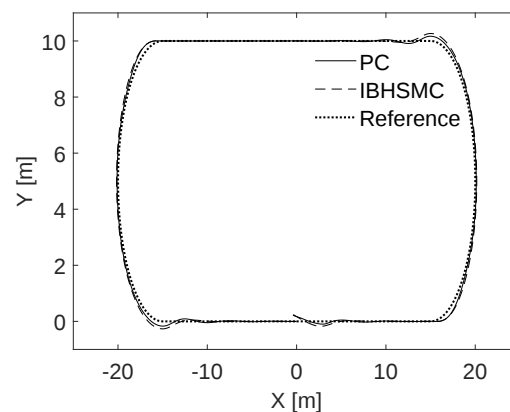


Figure 2. Simulation results on the trajectory tracking performances of the ball-balancing robot.

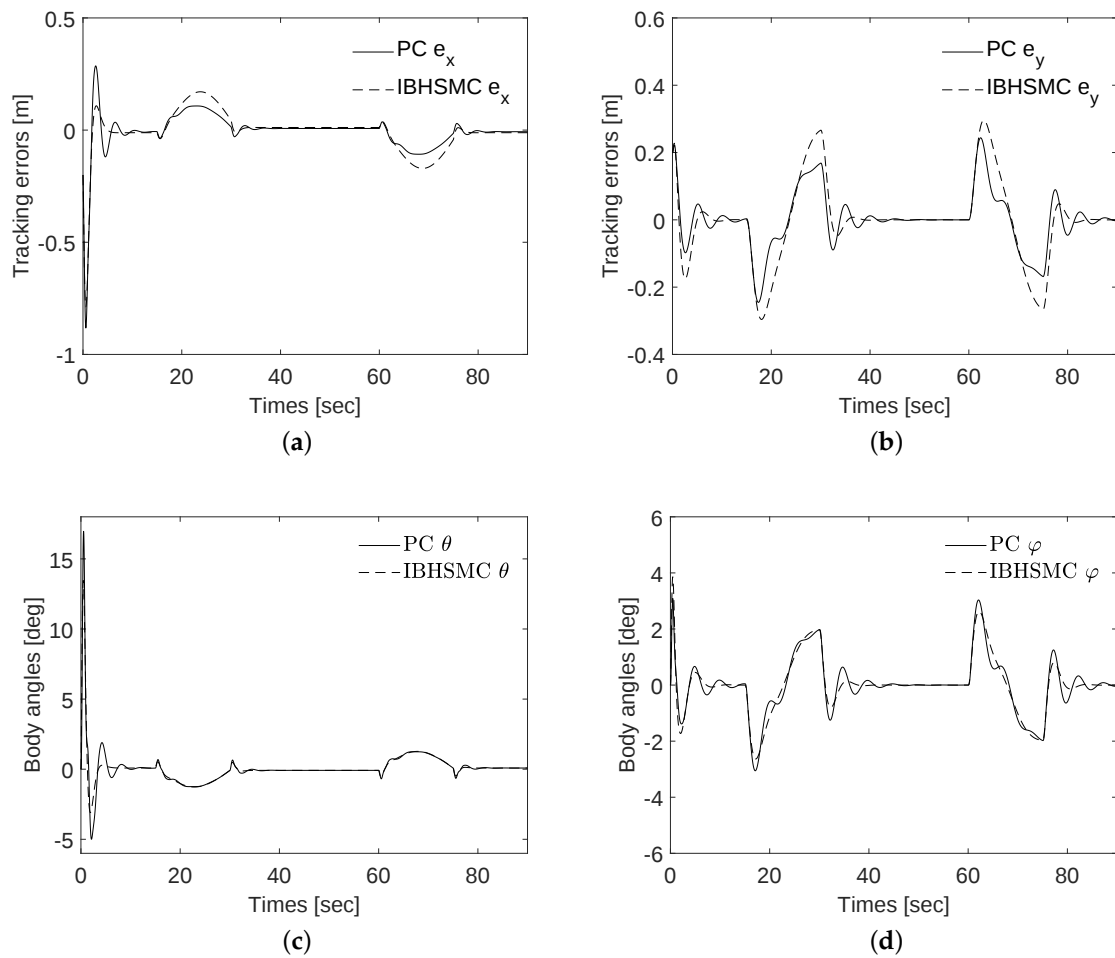


Figure 3. Simulation results (a) tracking error e_x ; (b) tracking error e_y ; (c) body angle θ ; (d) body angle φ .

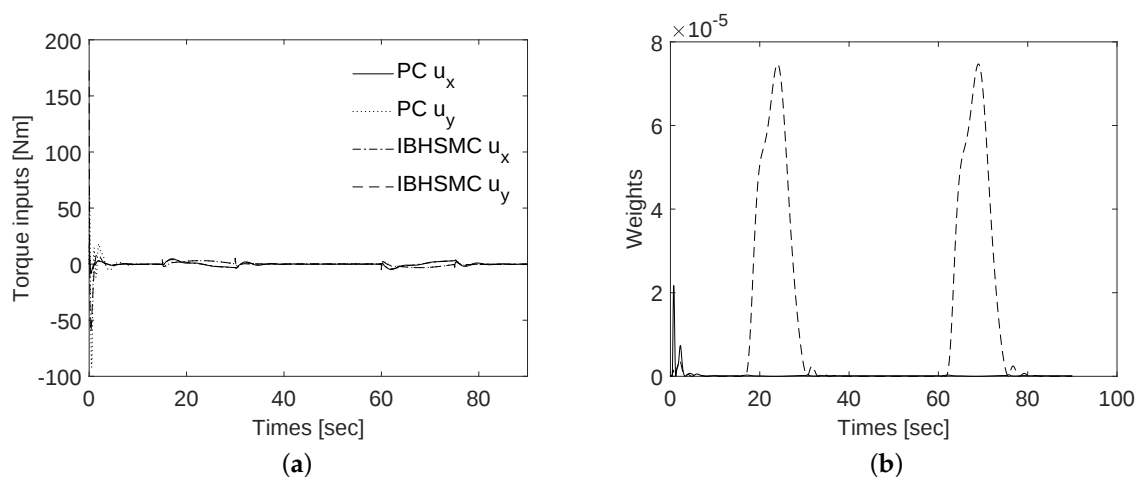


Figure 4. Simulation results (a) control inputs; (b) estimates of the optimal weights of the RBFNs (solid: $\|\hat{\omega}_1\|$, dotted: $\|\hat{\omega}_2\|$).

4. Discussion

In this paper, a neural-network-based control method was proposed for trajectory tracking and balancing of a ball-balancing robot. We introduced a virtual angle to solve the underactuation problem and designed a single-loop controller using the DSC method

for a simple control system. Moreover, we employed the RBFNs to compensate for the uncertainties and derived the update laws for the RBFNs in the Lyapunov sense. From the Lyapunov stability theorem, it was proven that all errors in the closed-loop control system were uniformly ultimately bounded and could be made arbitrarily small. The simulation results, including comparisons with the previous method, were provided to verify the performance of the proposed method. It was shown that the proposed method had a 19.2% improved tracking error rate compared to the previous method. In addition, it was confirmed that all errors could be made arbitrarily small as the choice of the design parameters shown in Table 2. Therefore, unlike other previous studies, the proposed method had a simple control system structure and guaranteed stability in achieving simultaneous trajectory tracking and balancing.

In this paper, it was assumed that all states, such as the position, body angle, and velocity of the robot, were measurable and exact. However, this may not be practical because it requires many sensors and can be contaminated by noise. The output-feedback control method using the observer can be a solution to this problem. Therefore, as future research, we will provide a control method that minimizes the use of sensors, such as output-feedback control.

Author Contributions: Conceptualization, H.-G.J. and B.-S.P.; methodology, H.J., C.-H.H., and B.-S.P.; software, H.-G.J.; validation, C.-H.H. and B.-S.P.; investigation, H.-G.J.; writing—original draft preparation, H.-G.J.; writing—review and editing, C.-H.H. and B.-S.P.; supervision, B.-S.P.; funding acquisition, B.-S.P. All authors read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government Grant NRF-2019A2C1087552.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The terms of the Lagrange equation are as follows [12]:

$$\begin{aligned}
 m_1 &= m_{bs} + 1.5(3S_\alpha^2 S_\varphi^2 - S_\alpha^2 + C_\varphi^2)I_{wr}, \\
 m_2 &= 0.75c_1 S_{2\varphi} S_\theta, \\
 m_3 &= 1.5(c_1 C_\varphi^2 - 2S_\varphi^2)r_s + lm_b C_\varphi C_\theta, \\
 m_4 &= -lm_b S_\varphi S_\theta, \\
 m_5 &= m_{bs} + 1.5(C_\alpha^2 C_\theta^2 + C_\alpha^2 S_\varphi^2 S_\theta^2 + 2C_\varphi^2 S_\alpha^2 S_\theta^2)I_{wr}, \\
 m_6 &= -1.5c_1 C_\varphi S_\varphi S_\theta I_{wr} r_s, \\
 m_7 &= 1.5r_s C_\alpha^2 C_\theta I_{wr} - lm_b C_\varphi, \\
 m_8 &= I_{by} C_\varphi^2 + m_b l^2 C_\varphi^2 + 1.5(3S_\alpha^2 S_\varphi^2 - S_\alpha^2 + C_\varphi^2)I_{wr} r_s, \\
 m_9 &= I_{bx} + m_b l^2 + 1.5r_s^2 C_\alpha^2 I_{wr}, \\
 c_{11} &= -3S_{2\varphi}(3C_{2\alpha} - 1)I_{wr} \dot{\varphi}/8, \\
 c_{12} &= 0.75c_1 I_{wr}(S_\theta C_{2\varphi} \dot{\varphi} + c_2 \dot{\theta}), \\
 c_{13} &= 0.75c_1 S_{2\varphi} I_{wr} \dot{x} + 0.75c_1 C_{2\varphi} S_\theta I_{wr} \dot{y} - lm_b C_\varphi S_\theta \dot{\varphi} - (1.5r_s C_\varphi c_1 - lm_b C_\theta)S_\varphi I_{wr} \dot{\theta}, \\
 c_{14} &= 0.75c_1 c_2 I_{wr} \dot{y} - (1.5c_1 I_{wr} r_s C_\varphi + lm_b C_\theta)S_\varphi \dot{\varphi} - lm_b C_\varphi S_\theta \dot{\theta}, \\
 c_{21} &= 0.75c_1 C_{2\varphi} S_\theta I_{wr} \dot{\varphi} + 0.75c_1 C_\varphi C_\theta S_\theta I_{wr} \dot{\theta}, \\
 c_{22} &= 1.5c_1 C_\varphi^2 C_\theta S_\theta I_{wr} \dot{\theta} - 1.5c_1 C_\varphi S_\theta^2 S_\varphi I_{wr} \dot{\varphi}, \\
 c_{23} &= 0.75c_1 C_{2\varphi} S_\theta I_{wr} \dot{x} - 0.75c_1 S_{2\varphi} S_\theta I_{wr} \dot{y} + lm_b S_\varphi \dot{\varphi} - 0.5(c_1 C_\varphi^2 + 2C_{2\varphi})r_s S_\theta I_{wr} \dot{\theta}, \\
 c_{24} &= 0.75c_1 C_\varphi C_\theta S_\varphi I_{wr} \dot{x} + 1.5c_1 C_\varphi^2 C_\theta S_\theta I_{wr} \dot{y} + (3 - 3c_1 C_\varphi^2 - 2C_\alpha^2)S_\theta I_{wr} r_s \dot{\varphi}/2 \\
 &\quad - 1.5c_1 c_2 I_{wr} r_s \dot{\theta},
 \end{aligned}
 \tag{A1}$$

$$\begin{aligned}
c_{31} &= 3(3C_{2\alpha} - 1)S_{2\varphi}C_{\theta}I_{wr}\dot{\varphi}/16 + 3(3C_{2\alpha} - 1)S_{2\varphi}I_{wr}r_s\dot{\varphi}/8, \\
c_{32} &= 3I_{wr}(c_3S_{\theta}r_s\dot{\varphi}/2 - c_1c_2\dot{x}/4 - c_1C_{\varphi}^2C_{\theta}S_{\theta}\dot{\varphi}/2), \\
c_{33} &= -0.75c_1S_{2\varphi}I_{wr}r_s\dot{x} + 1.5c_3S_{\theta}I_{wr}r_s\dot{\varphi} + (3r_s^2c_1 - 2m_b l^2 - 2I_{by})S_{2\varphi}I_{wr}\dot{\theta}/4, \\
c_{34} &= -(4m_b l^2 + 4I_{by} + 3(3C_{2\alpha} - 1)S_{2\varphi}I_{wr}r_s^2)\dot{\varphi}/8, \\
c_{41} &= -3c_1S_{2\varphi}I_{wr}\dot{x} - 3c_1S_{\theta}C_{2\varphi}I_{wr}\dot{\varphi} + 0.75c_1S_{2\varphi}I_{wr}r_s\dot{\theta}, \\
c_{42} &= 0.75I_{wr}(c_1S_{\theta}^2S_{2\varphi}\dot{\varphi} - c_1C_{2\varphi}S_{\theta}\dot{x} - 2c_3S_{\theta}r_s\dot{\theta}), \\
c_{44} &= 0.75c_1S_{2\varphi}r_sI_{wr}\dot{x} - 1.5c_3S_{\theta}r_sI_{wr}\dot{\varphi} + (2m_b l^2 - 3c_1I_{wr}r_s^2 + 4I_{by})S_{2\varphi}\dot{\theta}/4, \\
g_3 &= -glm_bC_{\varphi}S_{\theta}, \\
g_4 &= -glm_bC_{\theta}S_{\varphi},
\end{aligned}$$

where $C_{(\cdot)}$ and $S_{(\cdot)}$ denote $\cos(\cdot)$ and $\sin(\cdot)$, respectively, and $c_1 = (2 - 3C_{\alpha}^2)$, $c_2 = C_{\varphi}C_{\theta}S_{\varphi}$, $c_3 = (S_{\alpha}^2 - c_1C_{\varphi}^2)$, $I_{wr} = 3I_wC_{\alpha}^2/(2r_w^2)$, and $m_{bs} = m_b + m_s + I_s/r_s^2$.

Appendix B

The terms in (2) and (3) are defined as:

$$\begin{aligned}
F_s &= M_1^{-1}(C_{11}\dot{q}_s + C_{12}\dot{q}_b + g_1), \\
F_b &= M_2^{-1}(C_{21}\dot{q}_s + C_{22}\dot{q}_b + g_2), \\
\Delta F_s &= M_1^{-1}f_1, \Delta F_b = M_2^{-1}f_2, \\
G_s &= M_1^{-1}B_1, G_b = M_2^{-1}B_2, \\
M_1 &= M_{sb}^{-1}M_{ss}^{-1} - M_{bb}^{-1}M_{bs}^{-1}, \\
M_2 &= M_{ss}^{-1}M_{sb}^{-1} - M_{bs}^{-1}M_{bb}^{-1}, \\
C_{11} &= M_{bb}^{-1}C_{bs} - M_{sb}^{-1}C_{ss}, C_{12} = M_{bb}^{-1}C_{bb} - M_{sb}^{-1}C_{sb}, \\
C_{21} &= M_{bs}^{-1}C_{bs} - M_{ss}^{-1}C_{ss}, C_{22} = M_{bs}^{-1}C_{bb} - M_{ss}^{-1}C_{sb}, \\
f_1 &= M_{bb}^{-1}f_b - M_{sb}^{-1}f_s, f_2 = M_{bs}^{-1}f_b - M_{ss}^{-1}f_s, \\
g_1 &= M_{bb}^{-1}g_b, g_2 = M_{bs}^{-1}g_b, \\
B_1 &= M_{sb}^{-1}B_s - M_{bb}^{-1}B_b, B_2 = M_{ss}^{-1}B_s - M_{bs}^{-1}B_b, \\
M_{ss} &= \begin{bmatrix} m_1 & m_2 \\ m_2 & m_5 \end{bmatrix}, M_{bb} = \begin{bmatrix} m_8 & 0 \\ 0 & m_9 \end{bmatrix}, \\
M_{sb} &= M_{bs}^T = \begin{bmatrix} m_3 & m_4 \\ m_6 & m_7 \end{bmatrix}, \\
C_{ss} &= \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}, C_{sb} = \begin{bmatrix} c_{13} & c_{14} \\ c_{23} & c_{24} \end{bmatrix}, \\
C_{bs} &= \begin{bmatrix} c_{31} & c_{32} \\ c_{41} & c_{42} \end{bmatrix}, C_{bb} = \begin{bmatrix} c_{33} & c_{34} \\ 0 & c_{44} \end{bmatrix}, \\
f_s &= [\zeta_x\dot{x} \quad \zeta_y\dot{y}]^T, f_b = [\zeta_{\theta}\dot{\theta} \quad \zeta_{\varphi}\dot{\varphi}]^T, g_b = [g_3 \quad g_4]^T, \\
B_b &= \frac{1}{r_w} \begin{bmatrix} 0 & -C_{\varphi} \\ C_{\theta} & S_{\varphi}S_{\theta} \end{bmatrix}, B_s = \frac{r_s}{r_w} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.
\end{aligned}$$

References

1. Paszkiel, S. Using the Raspberry PI2 Module and the Brain-Computer Technology for Controlling a Mobile Vehicle. In *Automation 2019*; Szewczyk, R., Zieliński, C., Kaliczyńska, M., Eds.; Springer International Publishing: Cham, Switzerland, 2020; Volume 920, pp. 356–366.
2. Han, H.Y.; Han, T.Y.; Jo, H.S. Development of omnidirectional self-balancing robot. In Proceedings of the 2014 IEEE International Symposium on Robotics and Manufacturing Automation (ROMA), Kuala Lumpur, Malaysia, 15–16 December 2014; pp. 57–62.

3. Tsai, C.C.; Chan, C.K.; Kau, L.C. LQR Motion Control of a Ball Riding Robot. In Proceedings of the 2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Tokyo, Japan, 11–14 July 2012; pp. 861–866.
4. Lauwers, T.B.; Kantor, G.A.; Hollis, R.L. A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; pp. 2884–2889.
5. Nagarajan, U.; Kantor, G.; Hollis, R. The ballbot: An omnidirectional balancing mobile robot. *Int. J. Robot. Res.* **2014**, *33*, 917–930. [[CrossRef](#)]
6. Pham, D.B.; Weon, I.S.; Lee, S.G., Partial feedback linearization double-loop control for a pseudo-2D rideable ballbot. *Int. J. Control. Autom. Syst.* **2020**, *18*, 1310–1323. [[CrossRef](#)]
7. Pham, D.B.; Kim, H.; Kim, J.; Lee, S.G. Balancing and transferring control of a ball segway using a double-loop approach [applications of control]. *IEEE Control Syst. Mag.* **2018**, *38*, 15–37.
8. Wang, W.; Yi, J.; Zhao, D.; Liu, D., Design of a stable sliding-mode controller for a class of second-order underactuated systems. *IEE Proc.-Control. Theory Appl.* **2004**, *151*, 683–690. [[CrossRef](#)]
9. Pham, D.B.; Lee, S.G. Hierarchical sliding mode control for a two-dimensional ball segway that is a class of a second-order underactuated system. *J. Vib. Control* **2019**, *25*, 72–83. [[CrossRef](#)]
10. Pham, D.B.; Lee, S.G. Aggregated hierarchical sliding mode control for a spatial rideable ballbot. *Int. J. Precis. Eng.* **2018**, *19*, 1291–1302. [[CrossRef](#)]
11. Do, V.T.; Lee, S.G.; Kim J.H. Robust integral backstepping hierarchical sliding mode controller for a ballbot system. *Mech. Syst. Signal Process.* **2020**, *144*, 106866. [[CrossRef](#)]
12. Do, V.T.; Lee, S.G. Neural Integral Backstepping Hierarchical Sliding Mode Control for a Rideable Ballbot Under Uncertainties and Input Saturation. *IEEE Trans. Syst. Man, Cybern. Syst.* **2020**, 1–14. [[CrossRef](#)]
13. Park, K.B. Comments on ‘design of a stable sliding-mode controller for a class of second-order underactuated Systems’. *IET Control Theory Appl.* **2012**, *6*, 1153. [[CrossRef](#)]
14. Lee, S.M.; Park, B.S. Robust Control for Trajectory Tracking and Balancing of a Ballbot. *IEEE Access* **2020**, *8*, 159324–159330. [[CrossRef](#)]
15. Sands, T. Development of Deterministic Artificial Intelligence for Unmanned Underwater Vehicles (UUV). *J. Mar. Sci. Eng.* **2020**, *8*, 578. [[CrossRef](#)]
16. Fan, Y.; Huang, H.; Tan, Y. Robust adaptive path following control of an unmanned surface vessel subject to input saturation and uncertainties. *Appl. Sci.* **2019**, *9*, 1815. [[CrossRef](#)]
17. Sands, T. *Deterministic Artificial Intelligence*; IntechOpen: London, UK, 2020; ISBN 978-1-78984-111-4.
18. Tran, D.T.; Nguyen, M.N.; Ahn, K.K. RBF Neural Network Based Backstepping Control for an Electrohydraulic Elastic Manipulator. *Appl. Sci.* **2019**, *9*, 2237. [[CrossRef](#)]
19. Swaroop, D.; Hedrick, J.K.; Yip, P.P.; Gerdes, J.C. Dynamic surface control for a class of nonlinear systems. *IEEE Trans. Autom. Control* **2000**, *45*, 1893–1899. [[CrossRef](#)]
20. Ge, S.S.; Wang, C. Adaptive neural control of uncertain MIMO nonlinear systems. *IEEE Trans. Neural Netw.* **2004**, *15*, 674–692. [[CrossRef](#)] [[PubMed](#)]