actuators

MDPI

Article

Coordinated Trajectory Tracking and Self-Balancing Control for Unmanned Bicycle Robot Against Disturbances

Jinghao Liu [1], Chengcheng Dong [1], Xiaoying Lu [1], Qiaobin Liu [2] and Lu Yang [1,*]

[1] School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China; 3220240553@bit.edu.cn (J.L.); dcc9091457@163.com (C.D.); 1120223022@bit.edu.cn (X.L.)
[2] School of Mechanical & Automotive Engineering, South China University of Technology, Guangzhou 510641, China; liuqiaobin@scut.edu.cn
* Correspondence: yanglu@bit.edu.cn

Abstract

Trajectory tracking and self-balancing capacity is crucial for an unmanned bicycle robot (UBR) applied in off-road trails and narrow space. However, self-balancing is hard to be guaranteed once the steering angle manipulates for the tracking task, both of which are closely linked to the steering angle, especially for the UBR without auxiliary mechanism. In this paper, we introduce a double closed-loop framework in which the outer loop controller plans the desired speed and heading angle to track the reference trajectory, and the inner loop controller track the desired signals obtained from the outer loop to maintain balance. To be specific, a saturated velocity planner is developed to realize fast convergence of tracking error considering the kinematic constraints in the outer loop. A fuzzy sliding model controller (FSMC) is designed to attenuate the chattering effect via adapting its control gain in the inner loop, and a radial basis function neural network (RBFNN) approximator is also integrated into the framework to enhance the adaptability and robustness against bounded disturbances. The feasibility and effectiveness of the proposed control framework and approaches are validated based on the MATLAB and GAZEBO environment. In particular, the UBR can follow the testing route with lateral deviation less than 0.5 m in the presence of lateral winds and physical parameter measurement error, and comparative simulation results highlighted the superiority of the proposed control scheme.

Keywords: unmanned bicycle robot; self-balancing control; trajectory tracking; identification and compensation; double closed-loop control

Check for updates

Academic Editor: Keigo Watanabe

Received: 5 December 2025
Revised: 31 December 2025
Accepted: 7 January 2026
Published: 13 January 2026

Copyright: © 2026 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.

1. Introduction

Unmanned bicycle robots (UBRs) have attracted significant attention for detection, transport, and first response domains while reducing the exposure of human to threats and dangers. Due to the three-dimensional spatial rolling of the two wheels and the self-stability in specific speed range, the bicycle combines agility and maneuverability property. To exploit such property superiority, motion control of the UBR needs to be urgently captured, which brings a challenging avenue for actuation mechanism, self-balancing, and trajectory tracking control.

Due to the lack of direct torque in the rolling direction, several mechanisms have been designed which are different in various aspects such as the number and/or types of actuators. The most well-known mechanism is the flywheel, which can directly generate a rolling torque when accelerating [1]. To generate larger rolling torque and improve the

loading capacity, some scholars have adopted twin control moment gyros (CMGs) with opposing arrangement [2], which can generate superimposed rolling torque and offset vertical torque [3]. The earliest engineering application of such an idea can be traced back to 1903 when the Irish mechanical engineer Louis Brennan invented the single-track train by placing two large CMGs opposite each other in the carriage, and rotated them through electrical devices. The train weighing 22 tons successfully maintained balance while carrying more than 40 people. Furthermore, some added a heavy pendulum to the vehicle body, which can swing in the roll direction to maintain balance by adjusting the center of mass like cyclist [4,5]. However, the weight of the additional device is about 25% of the vehicle weight and deterioration in the maneuverability cannot be ignored [6]. Recently, Japan HONDA company proposes a riding assistance system with rear-wheel-swing mechanism, which enables balance at extremely low speeds [7]. However, this special mechanism equipped with a four-bar linkage remains heavy and insecure. To ensure safety and low cost, this paper studies the UBR without any auxiliary mechanism.

Different controllers have been proposed to deal with the complex and nonlinear UBR dynamics. For instance, a non-singular terminal sliding mode controller, capable of realizing finite-time convergence of the roll angle was proposed in [8]. For the MIMO System, reference [9] adopts integral sliding mode control with an improved reaching law, which can reduce chattering and smooth the control input. Reference [10] proposes a scheme that does not adjust the SMC structure or rely on nonlinear sliding manifolds. By optimizing the control law through disturbance estimation, this scheme mitigates chattering while ensuring system stability. Zhang [11] proposed a variable-gain LQR to keep balance while the speed is changing. In [12], a novel command-filtered adaptive control scheme for uncertain nonlinear systems with unknown external disturbances was proposed. This scheme adopts an asymmetric Bouc–Wen model to characterize the hysteretic nonlinearity, which significantly improves the accuracy. Kim [13] designed a GPR-based data-driven state feedback controller to achieve reference tracking of nonlinear discrete-time system. Considering the adverse effects by sensor error and actuator fault, an active fault-tolerant control based on the integration of SMC, fault detection, and fault estimation visa residual signals was proposed in [14]. Li [15] proposed a cascaded PID control method, which enhances the response speed and control precision. Xu [16] designed a variable universe fuzzy exponential rate reaching law sliding mode controller. Choi [17] proposed a controller using a deep reinforcement learning algorithm, which creates reward functions and neural networks, enabling UBR to maintain balance and reach the desired position through instructions. Kumar [18] derived a reduced-order method to enhance capacity of resisting disturbance. Sun [19] proposed an adaptive law, which estimates high-order dynamics to avoid robust terms, and the accuracy of compensation and position was improved. Most controllers depend on the simplified dynamic model which accuracy limits their performance. Therefore, some scholars directly identify balance state points from the dynamic phase diagram. Yu [20] revealed the inherent dynamics and balance characteristic of UBR and the potential application in self-balancing, drifting, and other aspects through the Steady-State Manifold theory.

Some trajectory tracking controller have been designed based on Ackerman kinematic model. Zhou [21] employed the Pure Pursuit (PP) algorithm with an appropriate look-ahead distance for prompt tracking of the desired trajectory. Yu [22] proposed an integrated tracking algorithm, which employs the PP algorithm when the positional deviation exceeds a threshold and switches to the Model Predictive Control within the threshold. He [23] designed a Gaussian process approach to adapt the control gains and hyperparameters through a learning algorithm for trajectory tracking. Yi [24] designed the expected trajectory of the internal subsystem and completed the tracking task of the external subsystem

based on the Gaussian process regression model predictive control method. For a class of underactuated systems, Fang [25] proposed a variable-gain controller with online trajectory that satisfies the output constraints, and integrated proportional–integral regulator into the trajectory to ensure the precise convergence of the actuated output.

To coordinate trajectory tracking and self-balancing with bounded disturbance, this study proposes a double closed-loop control framework for UBRs, in which a saturation function-based velocity planner and an FSMC-based balance controller combined with RBFNN-based approximator are designed. Compared to the existing stability control methods for UBRs, the contribution of this study can be marked as the following significant points.

1. A dynamic equilibrium model with system uncertain term is constructed to ensure physical interpretability, in which external lateral wind disturbance and internal parameter measurement error are incorporated.
2. A double closed-loop framework, composed of planner, controller and approximator, is proposed for coordinating trajectory tracking and self-balancing control, which considerably reduces the design complexity of control system.
3. A virtual and visualized bicycle simulation validation platform is developed in GAZEBO to intuitively demonstrate the effectiveness and advantages of the proposed approach with the uncertain system.

The paper is organized as follows: Section 2 presents the dynamic and kinematic models of the UBR, as well as the natural wind model. Section 3 addresses the design of the double closed-loop controller composed of RBFNN approximator, FSMC, and saturated velocity planner. In Section 4, the simulation results are illustrated. Finally, conclusions are drawn in Section 5.

2. System Modeling of UBR

In this section, the kinematic and dynamic models of the UBR, as well as the description of the lateral wind disturbance, are respectively presented.

2.1. Kinematic Model

It is impractical to establish a fully accurate and comprehensive bicycle model. Therefore, the following assumptions are firstly given, focused on the scenarios of light load, low speed, flat road, straight-line and small steering angle motion,:

(1) The deformation caused by the suspension is neglected. The UBR is divided into four rigid bodies: the front wheel, the rear wheel, the fork, and the frame.
(2) Tire deformation is ignored. The wheels are in point contact with the ground, where the contact points of the front and rear wheels with the ground are denoted as P_f and P_r, respectively.
(3) It is assumed that neither lateral nor longitudinal slip occurs in the wheels. Therefore, nonholonomic constraint exists at the wheel–ground contact points.
(4) The wheels remain in contact with the ground and the ground is horizontal. Hence, the pitching motion is not considered.

To charactize the motion of the UBR, two coordinate frames are introduced; see Figure 1. The axes X and Y of the ground fixed frame span the horizontal plane and its Z axis gives the vertical direction. The vehicle body is rotated around the line P_rP_t with the roll angle θ. Thus, the x and z axes span the plane of the vehicle body and the y axis is vertical to the plane. The origin of the frame is located at the center of mass O_c. Ignoring longitudinal slip and lateral deviation of the wheels, the vehicle rotates around instantaneous steering center point O with speed v, steering angle δ_s, and steering radius r,

where O_r and O_f denote the center of the rear and front wheels, respectively. Due to the fork not being vertical to the ground, the fork–ground intersection point P_t deviates from the line P_r, P_f with lateral offset $L_t \sin \delta_s$. The offset of a standard bicycle is small enough to be ignored in kinematic modeling, yet it should be included in dynamic modeling due to the caster effect.

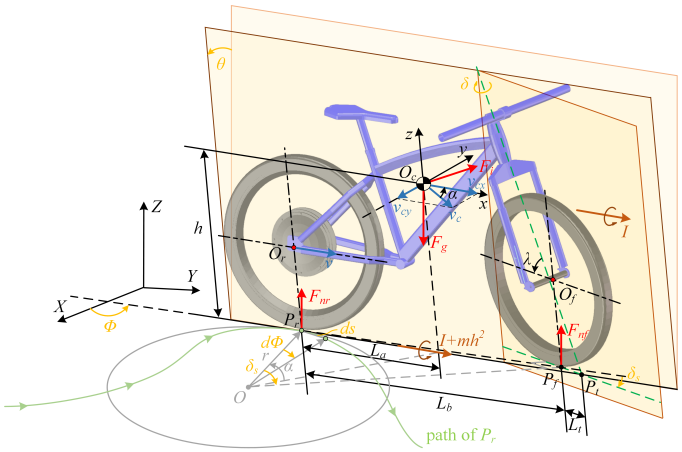


Figure 1. Schematic diagram of the UBR.

During the sample time dt, the distance traveled by the vehicle is ds and the change in the heading direction is $d\phi$. It yields

$$\frac{ds}{dt} = r\frac{d\phi}{dt} \tag{1}$$

where the turn radius r can be derived by geometric relationship:

$$r = \frac{L_a}{\tan \alpha} = \frac{L_b}{\tan \delta_s} \tag{2}$$

Combining Equations (1) and (2), the heading angular acceleration can be obtained:

$$\dot{\phi} = \frac{\tan \delta_s}{L_b}\frac{ds}{dt} = \frac{v \tan \delta_s}{L_b} \tag{3}$$

The speeds of the UBR in the ground fixed coordinate frame is

$$\begin{cases} v_x = v \cos \phi \\ v_y = v \sin \phi \end{cases} \tag{4}$$

Byassociating Equations (3) and (4), the kinematic model can be obtained as follows:

$$\begin{cases} \dot{x} = v \cos \phi \\ \dot{y} = v \sin \phi \\ \dot{\phi} = \frac{v \tan \delta_s}{L_b} \end{cases} \tag{5}$$

This equation can represent the position and heading angle of the UBR at any time.

The speed of central mass O_c is v_c, and v_{cx} and v_{cy} are the components of v_c along the x and y axis, respectively. Due to the frame being rigid, the equality $v_{cx} = v$ and relationships between v_c, v_{cy} and v can be obtained as follows:

$$v_c = r_c\dot{\phi} = \frac{L_a}{L_b}\frac{\tan \delta_s}{\sin \alpha}v \tag{6}$$

$$v_{cy} = v_c \sin\alpha = \frac{L_a \tan\delta_s}{L_b} v. \tag{7}$$

v_{cy} and its corresponding acceleration generate inertial forces along the y axis will be discussed in the next section.

2.2. Dynamic Model

It can be known that the UBR is subject to gravity F_g, support forces F_{nf} and F_{nr}, and inertial force F_i. Since the UBR always performs the variable speeds motion, the vehicle body coordinate frame is non-inertial at transient state; thus, it is necessary to introduce inertial force to analyze the balance. Meanwhile, Newton's laws are also applicable in this frame. According to the principle of torque balance, the equilibrium dynamic model can be obtained as follows:

$$(I + mh^2)\ddot{\theta} = F_g h \sin\theta + F_{nf} L_t \sin\delta_s + F_i h \cos\theta \tag{8}$$

where all terms are vectors and their directions follow the right-hand rule. I is the UBR's moment of inertia along the x axis. According to the parallel axis theorem, the moment of inertia along the line $P_r P_t$ is $I + mh^2$.

To clear understanding these torque, each torque term of the equation is specifically analyzed as follows:

(1) The first term is the gravitational torque, where $F_g = mg$.

$$F_g h \sin\theta = mgh \sin\theta \tag{9}$$

(2) The second term is the drag torque. As shown in Figure 1, the support forces are vertical to the ground and F_{nf} can be easily derived: $F_{nf} = \frac{L_a}{L_b} mg$. Then, the drag torque is

$$F_{nf} L_t \sin\delta_s = mg \frac{L_a L_t \sin\delta_s}{L_b} \tag{10}$$

(3) The third term is the inertial torque. The inertial force F_i is the sum of the inertial centrifugal force $m\frac{v^2}{r}$ due to circular motion and the inertial force $m\frac{dv_{cy}}{dt}$ caused by centripetal acceleration. This term is an important factor for balance. Then, follows that

$$F_i = m\left(\frac{dv_{cy}}{dt} + \frac{v^2}{r}\right) = m\left(\frac{L_a v \sec^2\delta_s}{L_b}\dot{\delta}_s + \frac{L_a \tan\delta_s}{L_b}\dot{v} + \frac{v^2}{L_b}\tan\delta_s\right) \tag{11}$$

Substituting Equations (9)–(11) into (8) obtains

$$\begin{aligned} c(I + mh^2)\ddot{\theta} = mgh\sin\theta + mg\frac{L_a L_t \sin\delta_s}{L_b} + \\ mh\cos\theta\left(\frac{L_a v \sec^2\delta_s}{L_b}\dot{\delta}_s + \frac{L_a \tan\delta_s}{L_b}\dot{v} + \frac{v^2}{L_b}\tan\delta_s\right) \end{aligned} \tag{12}$$

When the UBR is in straight-line or small-angle steering motion, the θ and δ_S are small. Therefore, Equation (12) can be simplified as follows:

$$(I + mh^2)\ddot{\theta} \approx mgh\theta + mg\frac{L_a L_t \delta_s}{L_b} + mh\left(\frac{L_a v}{L_b}\dot{\delta}_s + \frac{L_a \dot{v}}{L_b}\delta_s + \frac{v^2}{L_b}\delta_s\right) \tag{13}$$

Due to the fork angle λ, the relationship between the steering bar angle δ and the front-wheel steering angle δ_s is

$$\delta_s = \delta \sin\lambda \tag{14}$$

Substituting the above expression into (13) leads to

$$(I + mh^2)\ddot{\theta} = mgh\theta + mg\frac{L_a L_t \sin\lambda}{L_b}\delta + mh\left(\frac{L_a v \sin\lambda}{L_b}\dot{\delta} + \frac{L_a \dot{v} \sin\lambda}{L_b}\delta + \frac{v^2 \sin\lambda}{L_b}\delta\right) \quad (15)$$

2.3. Natural Wind Modeling

The external disturbance of natural wind cannot be ignored during the relatively big lateral area of UBR, especially when the wind force is strong. Natural wind has the characteristics of suddenness, persistence, periodicity, and uncertainty. Refer to the description of the wind in [26], the model can be achieved through the following methods.

The basic wind speed with a stable constant C is designed as follows:

$$V_a = C \quad 0 \le t \le t_a \quad (16)$$

The gust wind speed is designed as follows:

$$V_b = \begin{cases} 0 & t \le t_{b1}, t \ge t_{b2} \\ \frac{V_{\max 1}}{2}\left[1 - \cos 2\pi\left(\frac{t - t_{b1}}{t_{b1}}\right)\right] & t_{b1} < t < t_{b2} \end{cases} \quad (17)$$

where $V_{\max 1}$ is the peak of the gust, t_{b1} is the start time, and t_{b2} is the end time.

The gradual increase or decrease in the wind force can be described by the asymptotic wind speed:

$$V_c = \begin{cases} 0 & 0 \le t \le t_{c0},\ t > t_{c2} \\ V_{\max 2}\frac{t - t_{c0}}{t_{c1} - t_{c0}} & t_{c0} < t < t_{c1} \\ V_{\max 2} & t_{c1} \le t \le t_{c2} \end{cases} \quad (18)$$

where $V_{\max 2}$ is the peak speed, t_{c0} is the start time, t_{c1} is a time at a location, and t_{c2} is the end time.

The irregularity of the wind speed change can be described as follows:

$$V_d = V_{\max 3}\mathrm{Rand}(-1, 1)\cos(\omega t) \quad (19)$$

where $V_{\max 3}$ is the peak speed. $\mathrm{Rand}(-1, 1)$ generates random numbers that are uniformly distributed between -1 and 1. ω represents the average frequency.

The total wind speed is

$$V_w = V_a + V_b + V_c + V_d \quad (20)$$

Wind force can be expressed as

$$F_W = 0.5\rho S V_w^2 \quad (21)$$

where ρ denotes the density of air, the value is 1.29 kg/m^3, and S is the lateral windward area.

3. Double Closed-Loop Control System Design

In this section, we propose a double closed-loop control scheme for coordinating trajectory tracking and self-balancing of the UBR, as depicted in Figure 2.

Specifically, the outer control loop includes the saturated velocity planner based on the hyperbolic tangent function, which eliminates tracking errors and enhances stability under the saturation constraints of longitudinal velocity and yaw rate. Through inverse kinematic, the desired speeds can be transformed to roll angle and taken into the inner balance control loop, which integrates the RBFNN approximator and FSMC. By designing an appropriate fuzzy rule, FSMC adapts the exponential rate reaching law coefficient to

attenuate the chattering effect. The RBFNN approximator can identify and compensate for the disturbance term to further improve adaptability and robustness of the control system.

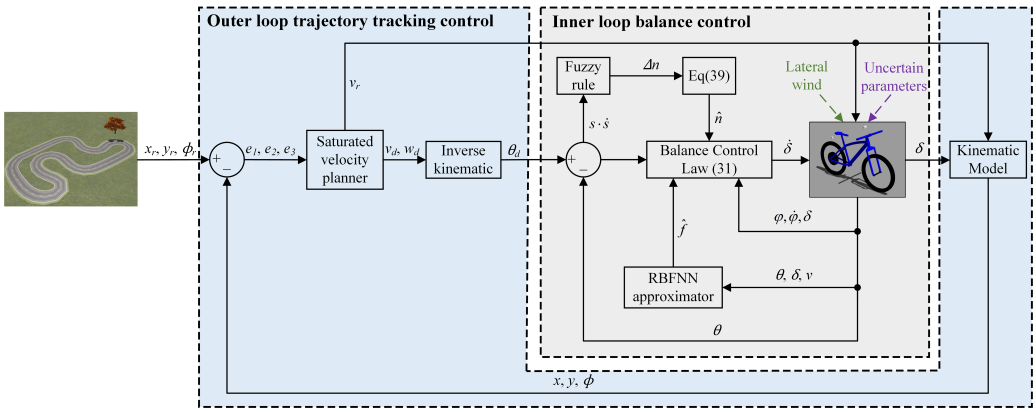


Figure 2. Double closed-loop control block diagram.

3.1. Sliding Mode Control Law Design

The lateral wind force can generate a torque $F_w h$. Therefore, the dynamic model with the disturbance term can be formulated as follows:

$$\begin{aligned}(I+mh^2)\ddot{\theta} &= mgh\theta + mg\frac{L_a L_t \sin\lambda}{L_b}\delta \\ +mh\left(\frac{L_a v \sin\lambda}{L_b}\dot{\delta} + \frac{L_a \dot{v} \sin\lambda}{L_b}\delta + \frac{v^2 \sin\lambda}{L_b}\delta\right) &+ F_w h\end{aligned} \tag{22}$$

To track the desired roll angle θ_d from the outer loop, the tracking error of the roll angle is formulated:

$$\begin{aligned} e &= \theta_d - \theta \\ \dot{e} &= \dot{\theta}_d - \dot{\theta} \\ \ddot{e} &= \ddot{\theta}_d - \ddot{\theta} \end{aligned} \tag{23}$$

Since the system is second-order, the sliding surface functions are defined as

$$\begin{aligned} s &= ce + \dot{e} \\ \dot{s} &= c\dot{e} + \ddot{e} \end{aligned} \tag{24}$$

where c must satisfy the Hurwitz condition, i.e., $c > 0$.

Substituting Equations (23) and (24) into (22), the expression of $\dot{s}$ can be obtained as follows:

$$\begin{aligned}\dot{s} = c\dot{e} + \ddot{\theta}_d - \frac{1}{I+mh^2}\left[mgh\theta + mg\frac{L_a L_t \sin\lambda}{L_b}\delta + mh\left(\frac{L_a v \sin\lambda}{L_b}\dot{\delta} + \frac{L_a \dot{v} \sin\lambda}{L_b}\delta + \frac{v^2 \sin\lambda}{L_b}\delta\right)\right] \\ - \frac{F_w h}{I+mh^2}\end{aligned} \tag{25}$$

To proof the stability of the controller, a Lyapunov function is selected as

$$\begin{aligned} V &= \frac{1}{2}s^2 \\ \dot{V} &= s\dot{s} \end{aligned} \tag{26}$$

To satisfy $s\dot{s} < 0$, the sliding mode control law u is obtained by putting the exponential rate reaching law $(\dot{s} = -n\,\mathrm{sgn}\,s - ks)$ into Equation (25):

$$\dot{\delta} = \frac{L_b(I+mh^2)}{L_a mhv\sin\lambda}\left\{c\dot{e} + \ddot{\theta}_d - \frac{1}{I+mh^2}\left[mgh\theta + mg\frac{L_aL_t\sin\lambda}{L_b}\delta + \frac{mh\delta\sin\lambda}{L_b}\left(L_a\dot{v}+v^2\right)\right] - \frac{F_w h}{I+mh^2} + n\,\mathrm{sgn}\,s + ks\right\} \tag{27}$$

Since the lateral wind force F_w is unknown, the sliding mode control law is

$$u = \dot{\delta} = \frac{L_b(I+mh^2)}{L_a mhv\sin\lambda}\left\{c\dot{e} + \ddot{\theta}_d - \frac{1}{I+mh^2}\left[mgh\theta + mg\frac{L_aL_t\sin\lambda}{L_b}\delta + \frac{mh\delta\sin\lambda}{L_b}\left(L_a\dot{v}+v^2\right)\right] + n\,\mathrm{sgn}\,s + ks\right\} \tag{28}$$

Then, substituting Equations (25) and (28) into (26) yields the following expression:

$$\begin{aligned}\dot{V} = s\dot{s} &= s(-n\,\mathrm{sgn}\,s - ks - \frac{F_w h}{I+mh^2}) \\ &= -n|s| - ks^2 - \frac{F_w h}{I+mh^2}s \le -n|s| \le 0\end{aligned} \tag{29}$$

where n denotes the upper bound of the disturbance, it needs to be sufficiently large, but a large upper bound will cause chattering. Since F_w may be negative due to changes in wind direction and $k > 0$, condition $n \ge \frac{|F_w|h}{I+mh^2}$ must be satisfied to ensure the above inequality.

Based on the above analysis, the Lyapunov function V is positive definite and $\dot{V}$ is negative definite. According to the LaSalle invariance principle, the control system is asymptotically stable. As $t \to \infty$, $s \to 0$ and the convergence rate of s depends on n.

3.2. RBFNN Approximator Design

The sliding mode control law (27) neglects the disturbance $F_w h$. We define the following disturbance term f to simplify the control law:

$$f = \frac{1}{I+mh^2}\left[mgh\theta + mg\frac{L_aL_t\sin\lambda}{L_b}\delta + \frac{mh\delta\sin\lambda}{L_b}\left(L_a\dot{v}+v^2\right)\right] + \frac{F_w h}{I+mh^2} \tag{30}$$

where f is uncertain under varying wind conditions and physical measurement error.

Substituting Equation (30) into (27) yields control law:

$$u = \frac{L_b(I+mh^2)}{L_a v\sin\lambda\cdot mh}\left(c\dot{e} + \ddot{\theta}_d - f + n\,\mathrm{sgn}\,s + ks\right) \tag{31}$$

To reduce its impact on self-balancing control, it employs the RBFNN to approximate the disturbance term f. The algorithm of this network is as follows:

$$h'_j = \exp\left(-\frac{\|x - c_j\|^2}{2b_j^2}\right) \tag{32}$$

$$f = W^{*T}h'(x) + \varepsilon \tag{33}$$

where x is the input vector, f is the output, i represents the i-th input of the network input layer, and j is the j-th input of the network hidden layer, $h' = [h'_j]^T$ is the output of the Gaussian basis function, W^* is the ideal network weight, ε is the network approximation error, and $\varepsilon \le \varepsilon_n$, c_j and b_j are, respectively, the center coordinates and width parameters of the Gaussian function.

Disturbances ultimately result in changes in θ and δ. Therefore, the network input is set as $x = [\theta, \delta]$, and the network output is

$$\hat{f}(x) = \hat{W}^T h'(x) \tag{34}$$

where $\hat{W}$ is the network weight designed by the adaptive law (35).

The obtained approximation term $\hat{f}$ replaces f in the control law u (31), and the new control law is

$$u = \dot{\delta} = \frac{L_b(I + mh^2)}{L_a mhv \sin\lambda}\left(c\dot{e} + \ddot{\theta}_d - \hat{f} + n\,\mathrm{sgn}\,s + ks\right) \tag{35}$$

Substituting Equations (30) and (35) into (25), we get

$$\dot{s} = -f + \hat{f} - \eta\,\mathrm{sgn}\,s - ks = -\tilde{f} - \eta\,\mathrm{sgn}\,s - ks \tag{36}$$

where

$$\tilde{f} = f - \hat{f} = W^{*T}h'(x) + \varepsilon - \hat{W}^T h'(x) = \tilde{W}^T h'(x) + \varepsilon, \tilde{W} = W^* - \hat{W}.$$

Define the Lyapunov function as ($\gamma > 0$), as follows:

$$L = \frac{1}{2}s^2 + \frac{1}{2}\gamma\tilde{W}^T\tilde{W} \tag{37}$$

From Equations (35) and (36), the following can be obtained:

$$\begin{aligned}\dot{L} &= s\dot{s} + \gamma\tilde{W}^T\dot{\tilde{W}} = s(-\tilde{f} - n\,\mathrm{sgn}\,s - ks) - \gamma\tilde{W}^T\dot{\hat{W}} \\ &= s(-\tilde{W}^T h'(x) - \varepsilon - n\,\mathrm{sgn}\,s - ks) - \gamma\tilde{W}^T\dot{\hat{W}} \\ &= -\tilde{W}^T\left[sh'(x) + \gamma\dot{\hat{W}}\right] - s(\varepsilon + n\,\mathrm{sgn}\,s + ks)\end{aligned} \tag{38}$$

The adaptive law is taken as

$$\dot{\hat{\mathbf{W}}} = -\frac{1}{\gamma}s\mathbf{h}'(x) \tag{39}$$

Substituting the adaptive law (39) into the Lyapunov function, we obtain

$$\dot{L} = -\varepsilon s - n|s| - ks^2 \tag{40}$$

Since the approximation error ε is very small, $\varepsilon \le \varepsilon_n$ and $k > 0$, it is necessary to satisfy $n \ge \varepsilon_n$ to ensure $\dot{L} \le 0$. The control system is asymptotically stable according to the LaSalle invariance principle.

3.3. FSMC Design

The inequality (29) indicates that the condition $n \ge \frac{|F_w|h}{I+mh^2}$ is necessary. However, a large n will cause chattering, since $-n\,\mathrm{sgn}\,s$ corresponds to the constant rate reaching term in the exponential reaching law. Given that the disturbance $\frac{|F_w|h}{I+mh^2}$ is time-varying, n also needs to be adjusted accordingly. To ensure the system rapidly reaches the sliding mode surface while reducing chattering, fuzzy control is introduced to update parameter n, and the fuzzy rules are designed based on experience as follows:

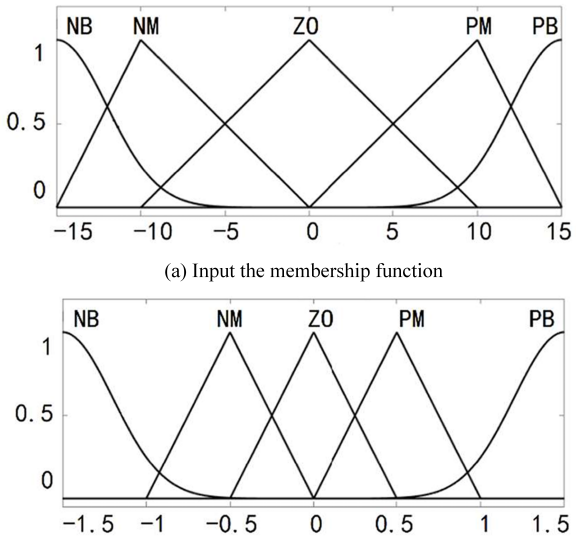
$$\begin{aligned}&\text{IF } s\cdot\dot{s} > 0, n \text{ should be increased} \\ &\text{IF } s\cdot\dot{s} < 0, n \text{ should be decreased}\end{aligned} \tag{41}$$

According to the fuzzy rule, the input variable is taken as $s \cdot \dot{s}$, and the output variable is taken as the switching gain parameter Δn. The definition of the input and output fuzzy is as follows:

$$\begin{aligned}s\dot{s} &= \{NB \quad NM \quad ZO \quad PM \quad PB\} \\ \Delta n &= \{NB \quad NM \quad ZO \quad PM \quad PB\}\end{aligned} \tag{42}$$

where NB is negative big, NM is negative medium, ZO is zero, PM is positive medium, and PB is positive big.

The membership functions of both input and output variables use triangular, Z-shaped and S-shaped functions, respectively, as shown in Figure 3.



(a) Input the membership function

(b) Output the membership function

Figure 3. The membership function curves of the input and output variables.

The fuzzy rules is designed as follows:

$$\begin{aligned} &\text{R1: IF } s\dot{s} \text{ is PB THEN } \Delta K \text{ is PB} \\ &\text{R2: IF } s\dot{s} \text{ is PM THEN } \Delta K \text{ is PM} \\ &\text{R3: IF } s\dot{s} \text{ is ZO THEN } \Delta K \text{ is ZO} \\ &\text{R4: IF } s\dot{s} \text{ is NM THEN } \Delta K \text{ is NM} \\ &\text{R5: IF } s\dot{s} \text{ is NB THEN } \Delta K \text{ is NB} \end{aligned} \tag{43}$$

The upper bound of the switching gain parameter n is estimated using the integral method:

$$\hat{n} = G \int_0^t \Delta n dt \tag{44}$$

where G is a proportional coefficient.

3.4. Saturated Velocity Planner Design

The state vector of the reference trajectory is $[x_r, y_r, \phi_r]$, and the actual state vector of the UBR obtained from the kinematic model is $[x, y, \phi]$. Therefore, the tracking error vector can be expressed as $[e_x, e_y, e_\phi] = [x_r - x,\ y_r - y,\ \phi_r - \phi]$. The longitudinal, lateral, and yaw angle error can be further obtained:

$$\begin{cases} e_1 = e_x \cos\phi + e_y \sin\phi \\ e_2 = e_y \cos\phi - e_x \sin\phi \\ e_3 = e_\phi \end{cases} \tag{45}$$

And the saturated velocity planner [27] is designed as follows:

$$\begin{cases} v_d = v_r \cos e_3 + \lambda_1 \tanh e_1 \\ \omega_d = \omega_r + \frac{\lambda_2 v_r e_2 \sin e_3}{(1+e_1^2+e_2^2)e_3} + \lambda_3 \tanh e_3 \end{cases} \tag{46}$$

where $\sin e_3/e_3 = \int_0^1 \cos(z \cdot e_3)\mathrm{d}z$, $v_r = \dot{x}_r \cos\phi_r + \dot{y}_r \sin\phi_r$ and $\omega_r = \dot{\phi}_r$ are the reference forward velocity and angular velocity of the trajectory, respectively. λ_1, λ_2 and λ_3 are the positive design parameters.

Consider the following Lyapunov function as

$$V_1 = \frac{\lambda_2}{2}\log(1 + e_1^2 + e_2^2) + \frac{1}{2}e_3^2 \tag{47}$$

Differentiating (47) with respect to time results in

$$\dot{V}_1 = \frac{\lambda_2(e_1\dot{e_1} + e_2\dot{e_2})}{1 + e_1^2 + e_2^2} + e_3\dot{e_3} \tag{48}$$

Differentiating (45) and incorporating the nonholonomic constraint $\dot{x}\sin\phi - \dot{y}\cos\phi = 0$ yields

$$\begin{cases} \dot{e}_1 = \omega e_2 - v + v_r \cos e_3, \\ \dot{e}_2 = -\omega e_1 + v_r \sin e_3, \\ \dot{e}_3 = \omega_r - \omega. \end{cases} \tag{49}$$

Substituting (46) and (49) into (48) yields the following result:

$$\dot{V}_1 = -\frac{\lambda_1\lambda_2 e_1 \tanh e_1}{1 + e_1^2 + e_2^2} - \lambda_3 e_3 \tanh e_3. \tag{50}$$

Considering the property of the hyperbolic tangent function, for any e_1 and e_3, we have $e_1 \tanh e_1 \geqslant 0$ and $e_3 \tanh e_3 \geqslant 0$. Therefore, for any initial error, it follows that $\dot{V}_1 \leqslant 0$ and $\lim_{t\to\infty}(|e_1(t)| + |e_2(t)| + |e_3(t)|) = 0$. Furthermore, according to Equation (45), we obtain $\lim_{t\to\infty}\left(|e_x(t)| + |e_y(t)| + |e_\varphi(t)|\right) = 0$.

Moreover, considering $|\tanh(\cdot)| < 1$, $\left|\frac{e_2}{1+e_1^2+e_2^2}\right| < 1$ and $\left|\frac{\sin e_3}{e_3}\right| < 1$, according to Equation (46) and the properties of inequalities, we have

$$\begin{aligned} |v_d| &\leqslant |v_r \cos e_3| + |\lambda_1 \tanh e_1| < |v_r| + \lambda_1, \\ |\omega_d| &\leqslant |\omega_r| + \left|\frac{\lambda_2 v_r e_2 \sin e_3}{(1 + e_1^2 + e_2^2)e_3}\right| + |\lambda_3 \tanh e_3| < |\omega_r| + \lambda_2|v_r| + \lambda_3. \end{aligned} \tag{51}$$

It can be seen from Equation (51) that by selecting different λ_1, λ_2, and λ_3, v_d and ω_d can have different ranges, thereby making the system meet the corresponding speed constraint.

To sum up, the double closed-loop control system can be realized by the following steps:

In the first step, the tracking error $[e_1, e_2, e_3]$ is obtained from Equation (45). Then, the forward velocity v_d and angular velocity ω_d required for tracking the trajectory can be obtained from the saturated velocity planner (46).

In the second step, the desired roll angle θ_d is obtained through inverse kinematic:

$$\theta_d = \arctan\left(-\frac{v_d w_d}{g}\right) \tag{52}$$

In the third step, θ_d is taken into the inner loop to realize the balance control.

The fourth step is to obtain actual state vector $[x, y, \phi]$ through the kinematic model (5) and take it to the outer loop to achieve closed-loop control.

Through the four steps, we build the double closed-loop control system to realize precise trajectory tracking and self-balancing simultaneously.

https://doi.org/10.3390/act15010049

4. Results and Discussions

In this section, we train the parameters and structure of the RBF neural network and then verify the above controller in MATLAB-SIMULINK and ROS-GAZEBO environment. Specifically, the study employs MATLAB R2023b, ROS Noetic, and GAZEBO 11.15 with the ROS-GAZEBO environment deployed on the Ubuntu 20.04 system.

4.1. RBF Neural Network Train

The physical parameters of the UBR are shown in Table 1. The location of the central mass is uniform random to simulate measurement error. Under normal trail condition, L_t is small, so its influence can be neglected.

Table 1. Structural parameters of the UBR.

Parameter Name	Symbol	Size (Unit)
Mass	m	165 (kg)
Front fork angle	λ	72°
Wheelbase	L_b	1.37 (m)
COG to rear axis	L_a	0.80∼0.85 (m)
COG's height	h	0.50∼0.52 (m)
The moment of inertia	I	11 (kg·m²)
Acceleration of gravity	g	9.8 (m/s²)

The simulation results of wind speed are presented in Figure 4, with the relevant parameters set as follows: $C = 3.4$ m/s, $V_{\max 1} = 3.0$ m/s, $V_{\max 2} = 2.0$ m/s, $V_{\max 3} = 2.0$ m/s, $t_a = 140$ s, $t_{b1} = 2$ s, $t_{b2} = 140$ s, $t_{c0} = 6$ s, $t_{c1} = 50$ s, $t_{c2} = 140$ s, $w = 2\pi$, and $S = 0.4$ m^2. The grade of wind varies from Level 1 to Level 5.

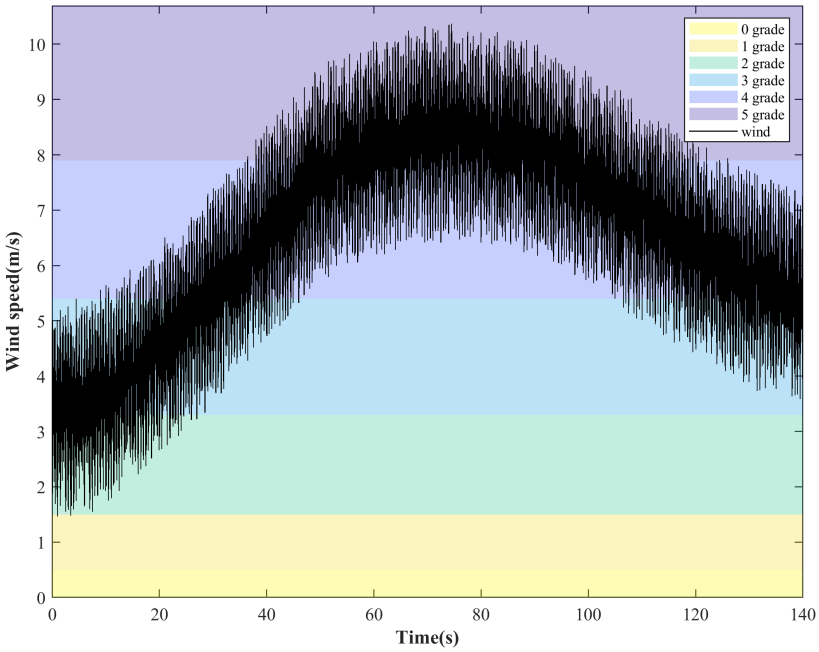


Figure 4. Wind speed and grade division.

During the train process, the inputs are the roll angle θ and the steering angle δ, and the output is the value of function f. We use 1000 random data as train samples with the input range $[-1.5, 1.5]$ and 500 random data as test samples. Since the actual inputs of the RBFNN are e and $\dot{e}$, the random input range of the test samples is set to $[-1, 1]$.

Furthermore, COG's height h and wind force F_w are also random to express the uncertainty. The training results are shown in Figures 5–7.

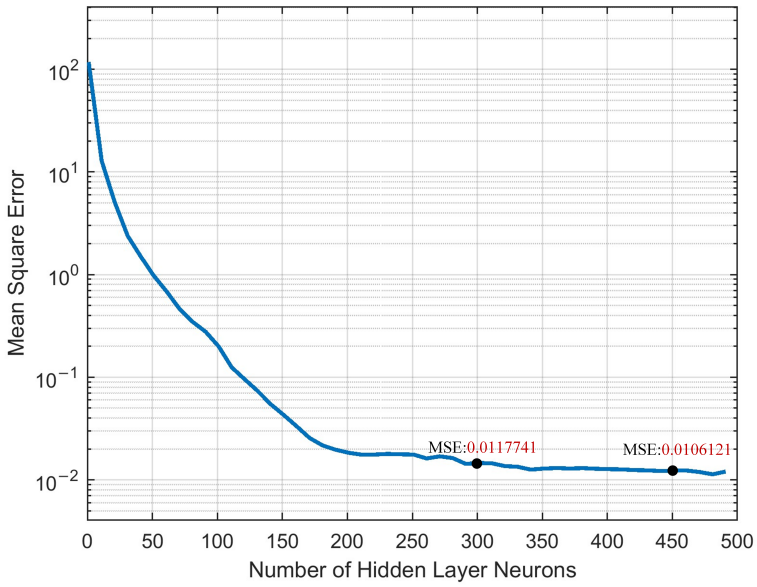


Figure 5. The curve of the mean square error (MSE) and the number of hidden layer neurons.

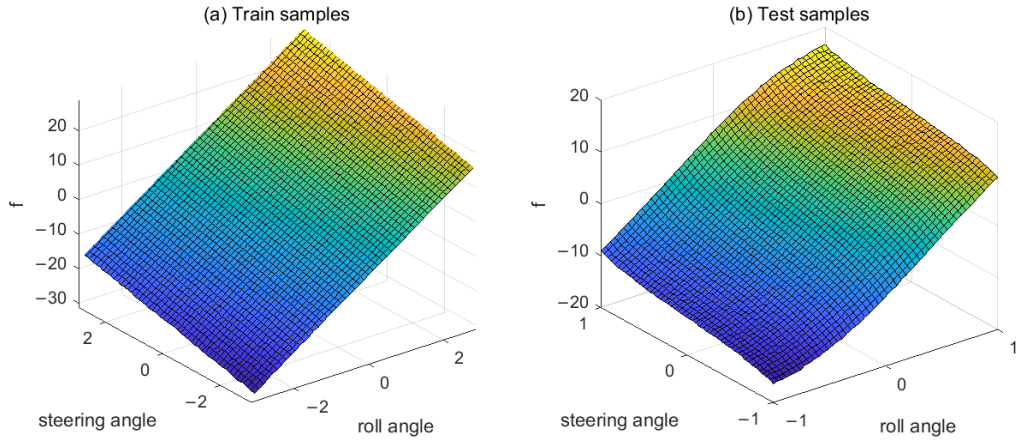


Figure 6. The curve of the train and test samples.

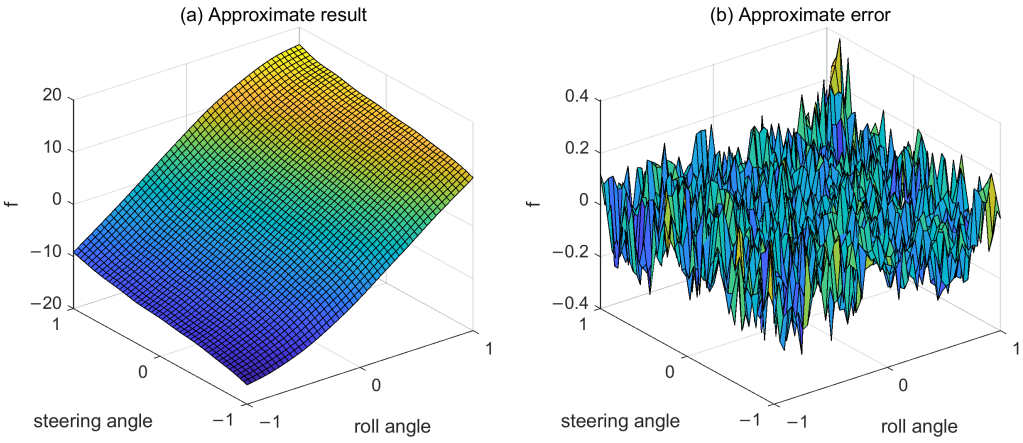


Figure 7. The curve of the approximate results and error.

It can be seen from Figure 5 that the mean square error (MSE) is 0.0117741 when the number of hidden layer neurons is 300. But the MSE only decreases by 0.001162 when the number is 450. Hence, we set the number of hidden layer neurons as 300 to avoid increasing the complexity of the network and ensure a good approximation effect. The structure of the RBFNN is 2-300-1 with the parameter $b_j = 0.65$ after training and optimization. The column size of matrix c_j is 2×300 related to the number of hidden layer neurons. It can be seen from Figure 7 that the optimized RBFNN has a good approximation effect on the test samples with errors distributed between $[-0.4, 0.4]$.

4.2. Matlab/Simulink

From the stability proofs of the FSMC in the above section, the ranges and implications of the control parameters have been derived. Specifically, these constraints include $c > 0$ and $n \geq \frac{|F_w|h}{I+mh^2}$, where n denotes the upper bound of the disturbance and must be selected as a sufficiently large value. However, a excessively large upper bound will cause chattering. Therefore, the control parameters of the FSMC should be carefully tuned in accordance with these constraints.

Moreover, with the forward velocity set as constant, λ_1 is taken as an extremely small value to make v_d almost equal to v_r based on the saturation constraints (51). In addition, λ_3 must be adjusted cautiously to maintain maneuverability while avoiding rollover caused by overly rapid heading adjustments. The final control parameters are shown in Table 2.

Table 2. Control parameters.

Controller Name	Symbol	Value
Planner	λ_1	0.000001
	λ_2	1.0
	λ_3	3.0
FSMC	c	50.0
	k	20.0
	n	10.0
	G	0.7
RBFNN	γ	20.0
	b_j	0.3

The reference trajectory data is provided by the Beijing Institute of Technology racing team, which includes straight segments and various curves. Fix the speed to $v = 2.0$ m/s and simulation step size to 0.01 s.

As illustrated in Figure 8, the UBR achieves precise tracking of the reference trajectory and always locates the outer side of the curves. As the reference path point in each computation cycle is chosen as the closest point ahead of the current rear wheel–ground contact point, a minor tracking delay is unavoidable, and this issue can be alleviated by decreasing the interval between reference path points. Specifically, Figure 9 shows that the trend of lateral deviation is similar to the curvature and their peak times are the same. The reason for the lag phenomenon is that the desired roll angle, the input to the balance control loop, is not a control quantity directly generated by the actuator, which causes the steering angle to rotate in the opposite direction to generate corresponding inertial force and, then, obtain desired roll angle. The phenomenon is also called "left-turn first-right". Although the lag cannot be avoided, the peak lateral deviation remains within 0.4 m and the peak yaw angle deviation does not exceed 4.5°. Therefore, the controller is effective in complex tracking scenario.

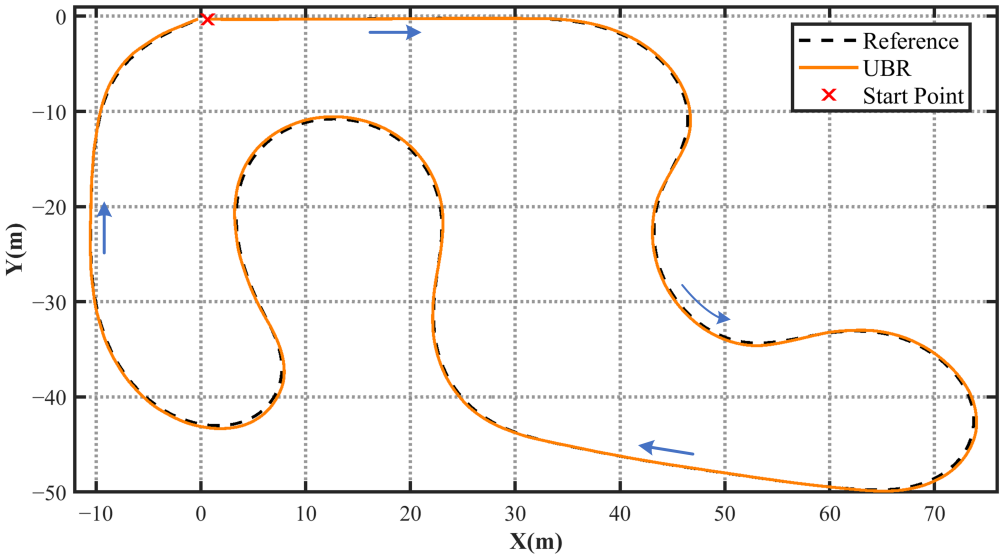


Figure 8. Desired versus UBR trajectory.

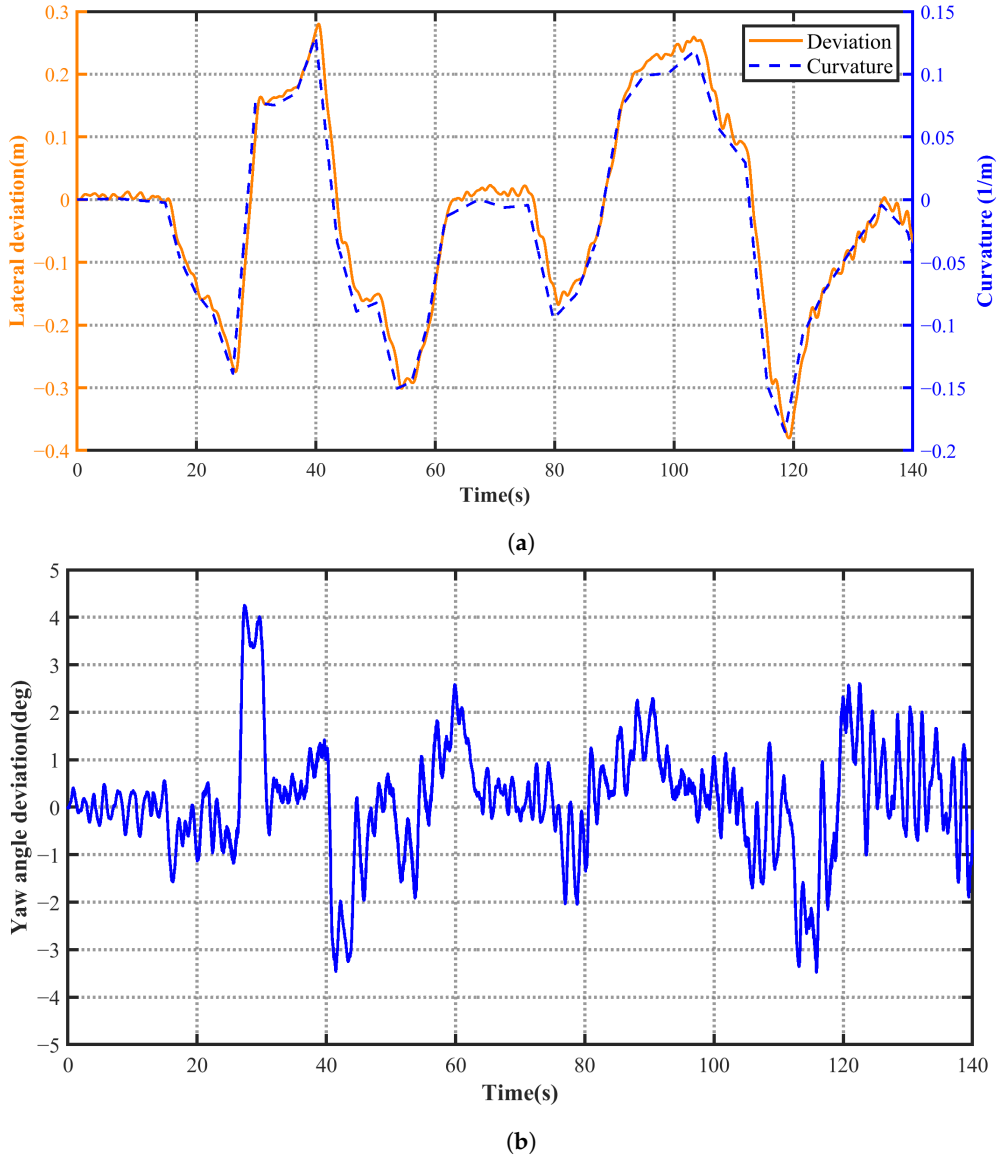


Figure 9. Numerical simulation to (**a**) lateral deviation and (**b**) yaw angle deviation.

https://doi.org/10.3390/act15010049

As can be seen from Figure 10, the trend of the steering angle and the roll angle is similar while their directions are opposite, indicating the coupling between these two variables. The simulation results of SMC and the proposed control law are compared under the same control parameters. It is known that both the chattering amplitude and frequency of the roll angle and handlebar angle are significantly reduced under the proposed law. Specifically, the chattering amplitude of the roll angle is constrained within 1°, the effect of which can be neglected. Meanwhile, the chattering amplitude of the handlebar angle is reduced by nearly 20°.

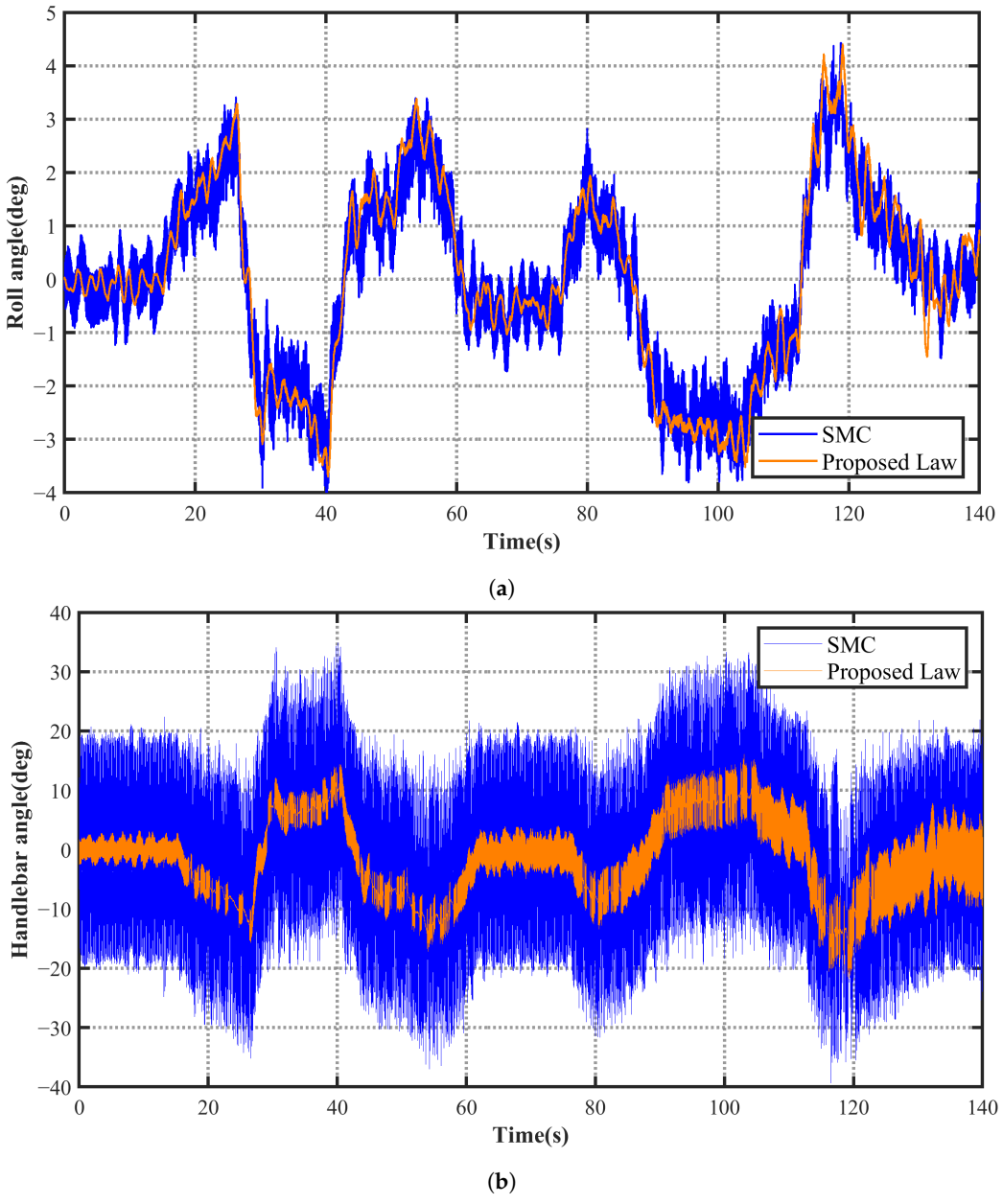


Figure 10. Numerical simulation to (**a**) roll angle and (**b**) handlebar angle.

The chattering frequency originates from the high-frequency random wind V_d. In the case of the SMC, such wind disturbance induces high-frequency fluctuations in the handlebar angle, which further lead the roll angle to chatter. In contrast, the proposed law is capable of approximating and compensating for the wind, thereby reducing the impact effectively.

The chattering amplitude is associated with the control parameter n. In the SMC, the coefficient n of the $n\mathrm{sgn}(s)$ term is fixed, which tends to cause large amplitude when the system state crosses the sliding surface. By contrast, the proposed law incorporates a fuzzy control to adaptively adjust this coefficient, thus effectively attenuating the chattering amplitude. What is more, the handlebar angle still rapidly varies due to the ignored steering dynamics. To address this, a low-pass filter is incorporated in the GAZEBO simulation to smooth the angle output, which can approximately simulate the effect of steering dynamics.

In summary, the roll angle is stably maintained within ±4.2°, and the lateral tracking deviation is maintained within 0.4 m, within the disturbances of wind and system parameter uncertainties. Therefore, it can be concluded that the UBR remains stable and balanced throughout the entire trajectory tracking process under the regulation of the proposed double closed-loop cooperative control scheme.

4.3. ROS/Gazebo

GAZEBO is a 3D dynamic simulator, enabling accurate and efficient simulation of UBR with physical properties. It offers high-fidelity physical simulations, sensor models, as well as user-friendly and program-friendly interaction interfaces [28].

First, a URDF file to describe the UBR is built, which contains precise physical information such as mass, inertia matrix, centroid position and so on. Inaccurate physical parameters could lead to unpredictable mistakes in simulation; thus, we adopted the open-source model from reference [29] and configured the controller based on it. The model is mainly made up of links and joints, while a joint connects two links through a parent–child relationship. The URDF model is shown in Figure 11.

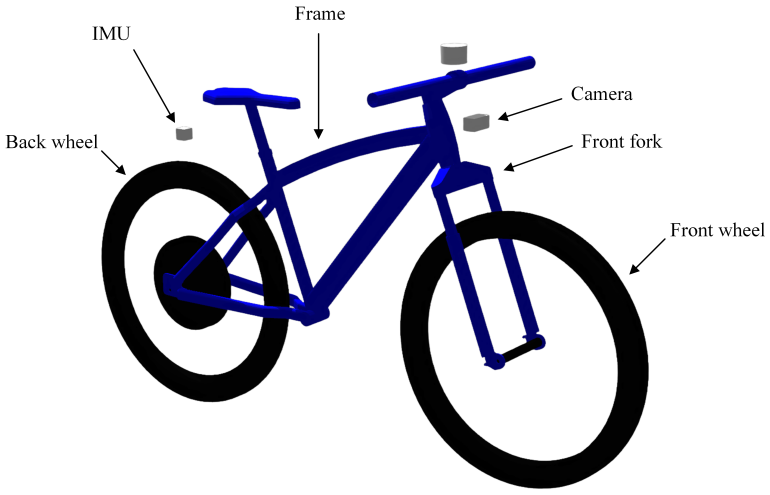


Figure 11. UBR model in GAZEBO.

After establishing the URDF model, it is necessary to configure motor controllers for driving the joints. The rear wheel rotation joint uses a velocity controller called "JointVelocityController", while the front fork steering joint employs a position controller called "JointPositionController". The underlying layer of these controllers employs the PID algorithm. Proper adjustment of the PID parameters is necessary for smooth joint movements, but it is quite complicated. To simplify this process, we smooth the control command instead. As shown in Equation (53), we first integrate the control output and then filter it. α is the smoothing coefficient, with $0 < \alpha < 1$. The closer α is to 1, the smoother the result, but the slower the response. In the experiment, α is set to 0.92. Finally, we describe the control type of each joint in the "yaml" file.

$$\begin{cases} \delta = \delta[t-1] + \dot{\delta} * dT \\ \delta[t] = \alpha * \delta[t-1] + (1-\alpha) * \delta \end{cases} \tag{53}$$

We used C++ 17 and Python 3.0 language to develop the nodes and topics for trajectory tracking and balance control of the UBR.

In this simulation, due to different physical parameters, the controller parameter λ_3 is modified to 2 while other parameters remain unchanged. The direction of the wind force is also set to be lateral at the centroid position. We use ros_bag tool to record the simulation results. The simulation video can be found at the following link: https://www.youtube.com/watch?v=-90zBuK3bcY (accessed on 31 December 2025).

By comparing the simulation results of MATLAB and GAZEBO as shown in Figure 12, the amplitude of lateral deviation in the GAZEBO simulation has increased by 0.2 m and the yaw deviation has increased by 6°. In Figure 13, the amplitude of roll angle has increased by 2° and the handlebar angle has decreased by 5° since the MATLAB does not consider steering dynamics. Such results are still within acceptable range, and the simulations in GAZEBO prove the feasibility of the proposed control law.

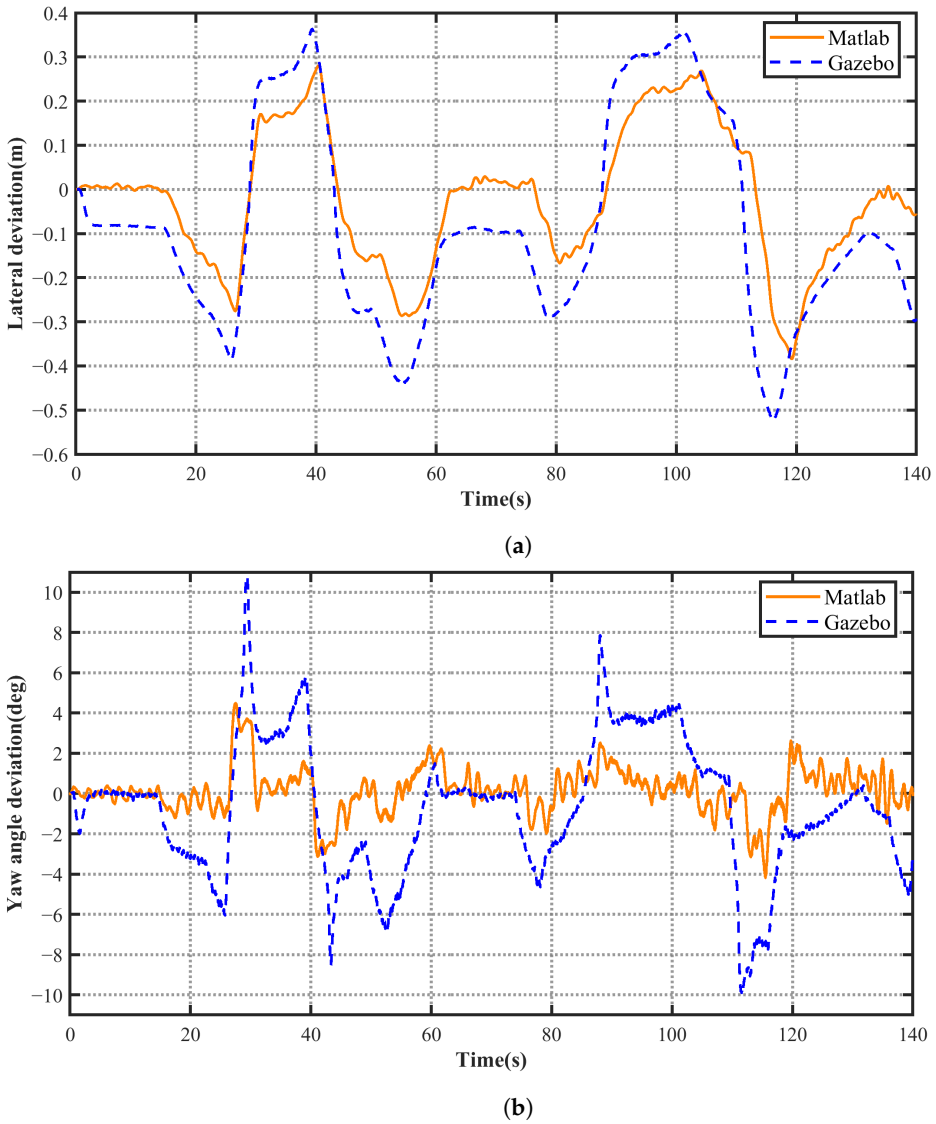
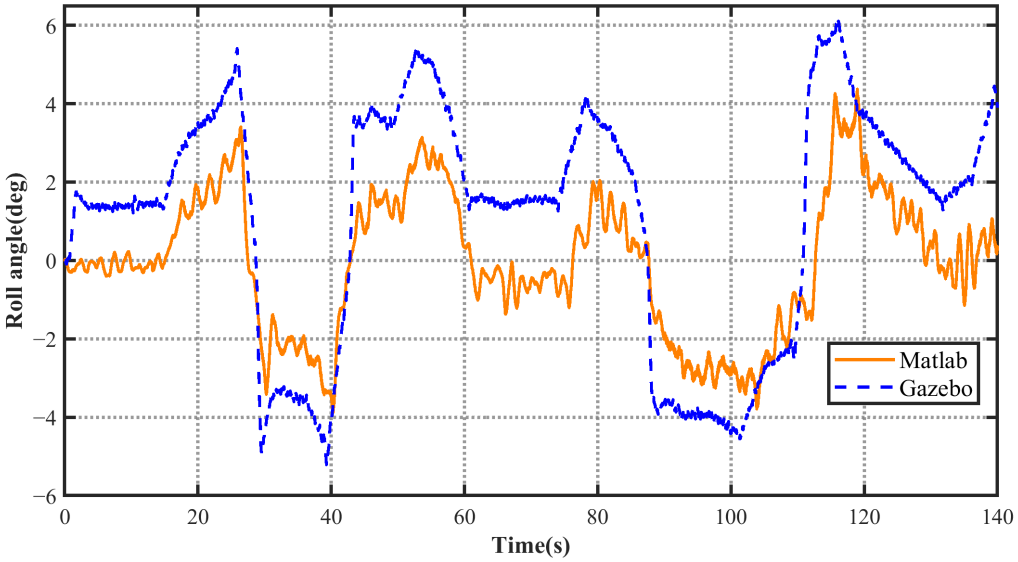
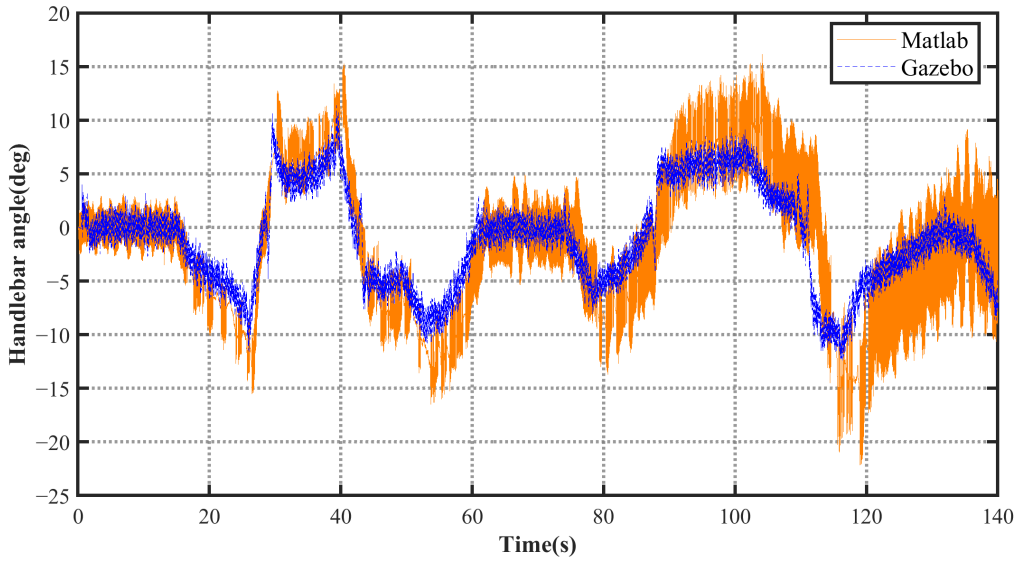


Figure 12. Comparative simulation experiment to (**a**) lateral deviation and (**b**) yaw angle deviation.



(**a**)



(**b**)

Figure 13. Comparative simulation experiment to (**a**) roll angle and (**b**) handlebar angle.

4.4. Discussions

Considering the future deployment of this control law on a real UBR, the potential challenges and corresponding countermeasures are summarized as follows:

1. In GAZEBO simulation, the precise position coordinates and roll angle can be directly obtained. However, the collected data are inevitably affected by noise and accuracy in real UBR. Therefore, it is necessary to use RTK technology to achieve centimeter-level location accuracy, and integrate the IMU with Kalman filter to obtain precise roll data.
2. It is different in the response speed and torque limitations of the actuator motor between the real world and the simulation, as well as the mechanical damping and stiffness. Therefore, it is essential to select motors and adjust control parameters to compensate for the differences.
3. A real UBR needs sufficient computing power to ensure real-time performance. For this purpose, the NVIDIA Jetson Orin NX equipped with the ROS system can be adopted. This main controller unit can not only provide adequate computing power, but also conveniently transplant simulation codes to the real UBR.

5. Conclusions

This study investigated coordinated self-balancing and trajectory tracking control of UBR against disturbances. We design RBFNN approximator to compensate disturbance term, and a fuzzy control is introduced to update the switching gain parameter of traditional SMC. The balance controller is integrated with a saturated velocity planner to construct a double closed-loop control system. Experimental validation performed in MATLAB–SIMULINK and ROS-GAZEBO environments confirmed the effectiveness and superiority of the proposed control scheme. Future research is directed toward integrating obstacle detection into the collision avoidance and autonomous navigation of the UBR.

Author Contributions: Methodology, J.L.; Validation, J.L., C.D. and X.L.; Formal analysis, L.Y.; Investigation, J.L. and X.L.; Writing—original draft, J.L.; Writing—review & editing, C.D. and L.Y.; Visualization, J.L.; Supervision, Q.L. and L.Y.; Project administration, X.L. and L.Y.; Funding acquisition, Q.L. and L.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by Open Foundation of National Key Laboratory of Multi-perch Vehicle Propulsion Systems (QDXT-NY-202407-08).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

1. Li, T.; Luo, P.; Gan, S.; Li, A. Balance and motion control of momentum wheeled bicycles. *J. Hunan Inst. Eng. (Nat. Sci. Ed.)* **2025**, *35*, 1–8.
2. Wang, B.; Chen, Z.; Zheng, Q.; Liu, X.; Zhao, M.; Liang, B. Observer-Based Disturbance Estimation and Optimal Allocation for the Roll Control of an Unmanned Motorcycle with Control Moment Gyros. In Proceedings of the IECON 2023—49th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 16–20 October 2023; pp. 1–6.
3. Zheng, X.; Zhu, X.; Chen, Z.; Sun, Y.; Liang, B.; Wang, T. Dynamic modeling of an unmanned motorcycle and combined balance control with both steering and double CMGs. *Mech. Mach. Theory* **2022**, *169*, 104643. [CrossRef]
4. Efraim, H.; Basri, R. Sensorless Unmanned Bicycle Robot. In Proceedings of the 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), Malé, Maldives, 16–18 November 2022; pp. 1–6.
5. Wang, B.; Deng, Y.; Chen, Z.; Liang, B. Equilibrium compensation based control: A universal control scheme for systems with mismatched disturbances and sensor errors. *IEEE Trans. Ind. Electron.* **2023**, *71*, 9590–9600. [CrossRef]
6. Araki, M.; Akimoto, K.; Takenaka, T. Study of riding assist control enabling self-standing in stationary state. *SAE Int. J. Veh. Dyn. Stab. NVH* **2018**, *3*, 47–56. [CrossRef]
7. Sumioka, T.; Akimoto, K.; Tsujimura, T.; Takayanagi, S.; Fukushima, K.; Nose, T. Rider cooperative control of rear-wheel-swing motorcycle based on divergent component of motion. *IEEE Robot. Autom. Lett.* **2023**, *9*, 223–230. [CrossRef]
8. Huang, Y.; Ren, Y.; Zhuang, W.; Luo, Z.; Liang, Z.; Hou, D. Research on Adaptive Non-singular Terminal Sliding Mode Balance Control for Unmanned Bicycles. *Meas. Control Technol.* **2024**, *43*, 37–48+75.
9. Irfan, S.; Zhao, L.; Ullah, S.; Javaid, U.; Iqbal, J. Differentiator- and Observer-Based Feedback Linearized Advanced Nonlinear Control Strategies for an Unmanned Aerial Vehicle System. *Drones* **2024**, *8*, 527. [CrossRef]
10. Ramesh, S.; Seo, D.; Drakunov, S. Correction: Sliding Mode Control with Chattering Reduction (American Institute of Aeronautics and Astronautics Inc., AIAA). In Proceedings of the AIAA Science and Technology Forum and Exposition, AIAA SciTech Forum 2025, Orlando, FL, USA, 6–10 January 2025.
11. Yongli, Z.; Liu, Y.; Yi, G. Model Analysis of Unmanned Bicycle and Variable Gain LQR Control. In Proceedings of the 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 11–13 Decembe 2020; Volume 9, pp. 1196–1200.
12. Zhang, Y.; Yao, Q.; Luo, B.; Chen, N. Robust control of uncertain asymmetric hysteretic nonlinear systems with adaptive neural network disturbance observer. *Appl. Soft Comput.* **2024**, *167*, 112387. [CrossRef]

13. Kim, H.; Chang, H. Model Reference Gaussian Process Regression: Data-Driven State Feedback Controller. *IEEE Access* **2023**, *11*, 134374–134381. [CrossRef]
14. Alizadeh, M.; Ramezani, A.; Saadatinezhad, H. Fault Tolerant Control in an Unmanned Bicycle Robot via Sliding Mode Theory. *Intell. Syst. Robot. (Engl. Ed.)* **2022**, *4*, 139–152. [CrossRef]
15. Li, F. Research on self-balancing unmanned bicycle based on cascade PID control. In Proceedings of the 2021 IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), Chongqing, China, 17–19 December 2021; Volume 2, pp. 442–446.
16. Xu, Z.; Fan, L.; Zhang, Y.; Li, H. Obstacle avoidance control of the unmanned bicycle based on variable universe fuzzy exponential rate reaching law sliding mode control. In Proceedings of the 2021 8th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS), Beijing, China, 10–12 December 2021; pp. 115–120.
17. Choi, S.; Le, T.P.; Nguyen, Q.D.; Layek, M.A.; Lee, S.; Chung, T. Toward Self-Driving Bicycles Using State-of-the-Art Deep Reinforcement Learning Algorithms. *Symmetry* **2019**, *11*, 290. [CrossRef]
18. Kumar, S.S.; Kumar, A. Modeliranje smanjenog reda i kontrola balansiranja biciklističkog robota. *FME Trans.* **2021**, *49*, 919–934. [CrossRef]
19. Yang, T.; Zhang, M.; Sun, W.; Sun, N. Fully-Actuated System Approach-Based Neuroadaptive Control for Underactuated Robots with State Estimation and Delay. *IEEE Trans. Cybern.* **2025**, *55*, 3234–3250. [CrossRef] [PubMed]
20. Tian, Y.; Chen, Z.; Deng, Y.; Wang, B.; Liang, B. Steady-State Manifold of Riderless Motorcycles. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 3491–3496.
21. Zhou, X.; Shi, X.; Huang, J. Closed-Loop Control for Self-Balancing and Steering of Non-Coaxial Two-Wheeled Robots. *Comput. Meas. Control* **2023**, *31*, 140–148.
22. Yu, Y.; Zhao, M. An Improved Path Following Algorithm for Autonomous Bicycle. *J. Huazhong Univ. Sci. Technol. (Nat. Sci. Ed.)* **2015**, *43*, 345–350.
23. He, K.; Deng, Y.; Wang, G.; Sun, X.; Sun, Y.; Chen, Z. Learning-Based Trajectory Tracking and Balance Control for Bicycle Robots with a Pendulum: A Gaussian Process Approach. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 634–644. [CrossRef]
24. Chen, K.; Yi, J.; Song, D. Gaussian-Process-Based Control of Underactuated Balance Robots with Guaranteed Performance. *IEEE Trans. Robot.* **2023**, *39*, 572–589. [CrossRef]
25. Zhao, Y.; Lu, B.; Fang, Y.; Zhi, J. Variable-Gain Coupling Online Trajectory for Underactuated Mechanical Systems with Unactuated Output Constraints. *IEEE Trans. Ind. Electron.* **2025**, *72*, 11865–11874. [CrossRef]
26. Yuan, R.; Zhang, Y.; Geng, L.; Li, X. Robust control of two-wheeled bicycle robot under wind disturbance. In Proceedings of the 2024 2nd Asia Conference on Electronics Engineering (ACEE), Singapore, 19–21 April 2024; pp. 23–29.
27. Yang, L.; Yue, M.; Liu, Y.; Guo, L. RBFNN based terminal sliding mode adaptive control for electric ground vehicles after tire blowout on expressway. *Appl. Soft Comput.* **2020**, *92*, 106304. [CrossRef]
28. Zhang, B.; Liu, P. Control and benchmarking of a 7-DOF robotic arm using Gazebo and ROS. *PeerJ Comput. Sci.* **2021**, *7*, e383. [CrossRef]
29. Yeh, T.J.; Lin, T.C.; Chen, A.C.B. Robust Balancing and Trajectory Control of a Self-Driving Bicycle. *IEEE Trans. Control Syst. Technol.* **2024**, *32*, 2410–2417. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.