

Research Article

Design Low-Order Robust Controller for Self-Balancing Two-Wheel Vehicle

Ngoc Kien Vu and Hong Quang Nguyen 

Thai Nguyen University of Technology, 666, 3/2 Street, Tich Luong Ward, Thai Nguyen City 251750, Vietnam

Correspondence should be addressed to Hong Quang Nguyen; quang.nguyenhong@tnut.edu.vn

Received 11 December 2020; Revised 26 April 2021; Accepted 11 May 2021; Published 24 May 2021

Academic Editor: Carlos Llopis-Albert

Copyright © 2021 Ngoc Kien Vu and Hong Quang Nguyen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

When there is no driver, balancing the two-wheel vehicle is a challenging but fascinating problem. There are various solutions for maintaining the balance of a two-wheel vehicle. This article presents a solution for balancing a two-wheel vehicle using a flywheel according to the inverted pendulum principle. Since uncertainties influence the actual operating environment of the vehicle, we have designed a robust controller RH_{∞} to maintain the vehicle equilibrium. Robust controllers often have a high order that can affect the actual control performance; therefore, order reduction algorithms are proposed. Using Matlab/Simulink, we compared the performance of the control system with different reduced-order controllers to choose a suitable low-order controller. Finally, experimental results using a low-order robust controller show that the vehicle balances steadily in different scenarios: no-load, variable load, stationary, and moving.

1. Introduction

Bicycles have appeared in the world for centuries as a popular means of transport. Two-wheel vehicles (including bicycles, motorbikes, and electric bicycles) have the advantage of low fuel consumption, fast acceleration, flexibility, narrow parking space, low cost, and being particularly suitable for limited road conditions in large cities. The most significant limitation of two-wheel vehicles is not being able to balance themselves like a car. Driving two-wheel vehicles is not as comfortable as driving a car. Besides, when colliding with another vehicle, the standard two-wheel vehicle will fall and cause injury to the driver. Therefore, the need to develop a two-wheeled vehicle that can balance itself both when stationary, in motion, and in a collision, creating comfort for the driver like when driving a car, is essential. If well designed, the self-balancing two-wheel vehicle will only drop off in the collision and remain vertical thanks to the self-balancing system mounted on it, ensuring the user's safety. However, the problematic issue is how to control the system

so that the system can balance itself under different working conditions, and the load carried can be changed simultaneously. Because the attractiveness of self-balancing two-wheel vehicles comes from theoretical and practical problems, research on self-balancing two-wheel vehicles has attracted the attention of many scientists and manufacturing companies.

In Vietnam today, two-wheel vehicles are still used more widely than cars. Therefore, the development of self-balancing two-wheel vehicles is essential and suitable for the traffic situation of Vietnam. This article builds a model to study and experimentally control the balance of the self-balancing two-wheel vehicle. This study will contribute to determining an appropriate model and algorithm for maintaining the balance of two-wheeled vehicles and then apply them into practice.

To be able to control the vehicle's balance, we must understand its dynamics. The bicycle model is nonintuitive and strongly depends on the speed [1]. When the vehicle is not in motion, it is an unstable model like an inverted

pendulum [1]. However, when the vehicle is in constant motion under certain conditions, it can be stable [1]. In [1], the authors mention that two-wheel vehicles can be described by various models with different complexity from the second-order linear model to fourth-order linear and nonlinear models. There are multiple methods to solve a two-wheel vehicle's balance problem without a driver, divided into two groups: with and without a stabilizer.

With the group without a stabilizer [2–6], to maintain the equilibrium state of the vehicle, the control system controls the vehicle's steering angle using an electric motor. This method is built on the nature of the bicycle when there is a driver. When the bicycle moves in the direction of control, it will generate a centrifugal force to maintain the equilibrium. Therefore, this method is very difficult to maintain the equilibrium when the vehicle is stationary.

With the method group with stabilizers, there are many solutions to design stabilizers to maintain vehicle equilibrium. In studies [7–14], the flywheel was energized by allowing it to rotate at high speed, and the control system controls the flywheel's tilt angle to generate force according to the principle of the gyroscope to maintain vehicle balance. The more the flywheel number, the faster the flywheel rotation speed, the greater the accumulated energy of the flywheel, and the better balance the vehicle. However, the number of flywheels used is usually one flywheel in [7–11] or two flywheels as in [12–14]. The advantage of this approach is that it responds quickly; the force to maintain the balance is significant, but the downside is that the flywheel rotates at high speed, so it dissipates a massive amount of energy.

In [15], the author added weight to the vehicle model and changed the position of the weight to change the vehicle's center of gravity to maintain equilibrium. The advantage of this method is that it is possible to keep the vehicle's balance when the vehicle is stationary and when the vehicle is in motion. However, to ensure a large balance force, the added weight is also large, leading to an increase in the size and weight of the vehicle, and the response of the system is often slow.

To overcome the disadvantage of the study in [15], the studies in [16, 17] suggested using flywheel-type weights to change the CoG. Accordingly, a flywheel will be installed on the vehicle. The flywheel structure can be altered to change its CoG, thereby changing the CoG of the vehicle.

The stabilizer using flywheel according to the principle of an inverted pendulum is proposed in [18–23]. In this method, the rotation direction of the flywheel will be changed to create a force that balances the gravity of the vehicle to maintain the equilibrium state. The advantage of this method is that the flywheel usually rotates at a low speed, so it dissipates less energy, and the response of the system is fast. The disadvantage of this method is that the balancing force is not large, so the vehicle can only balance within a small angle range.

We see that each group has its advantages and disadvantages. Some studies are looking to combine these methods, as specific as combining steering angle control with changing the CoG [24]. Through analysis of two-wheel vehicle balance control solutions, this article chooses the

technique using a flywheel to build a self-balancing two-wheel vehicle.

To control the self-balancing two-wheel vehicle according to the inverted pendulum principle, there are various methods such as PD [20], PID [17–19], LQG and MPC [21], SMC [25–27], and robust control [11, 28–38]. However, when a two-wheel vehicle works in reality, it will suffer many uncertain effects such as load, noise, and external force. Therefore, the two-wheel vehicle model can be considered as an uncertain object [10]. To control unstable objects, there are many different algorithms, of which the most used algorithm is sliding-mode control (SMC) [25–27] and robust control algorithm [11, 28–38]. Key features of the SMC are its ability to resist parameter uncertainty, a fast response rate, no sensitivity to bounded external disturbances, and simple design computation. The SMC control algorithm has been proven to control the balance of two-wheel vehicles according to the inverted pendulum model in [25]. SMC was improved and converted to NFTSM (non-singular fast terminal sliding-mode control) in [26] to better meet the requirements of robust control under the presence of uncertainties and disturbances. Robust finite-time composite nonlinear feedback control can be used to control uncertainties, as shown in [27].

The robust optimal control algorithm RH \ddot{Y} [11, 28–38] is designed to shape the system's response to achieve the desired behavior and maintain this behavior from model parameter variation or disturbances. However, to ensure the two above functions, the robust optimal controller H \ddot{Y} is often high. The high-order controller can make the controller's calculation time longer, hence the response speed of the controller slower, reducing the practical applicability of the controller. Therefore, the design of a low-order robust controller is a very concerning issue in robust control. To design a robust low-order controller, there are usually two basic solutions. In Solution 1, a low-order control structure with unknown parameters is selected; then, based on the robust optimization controller design criteria, these parameters are determined [11]. In Solution 2, a high-order stable controller is designed; then, the model order reduction algorithm is applied to reduce the controller [34, 38]. To implement Solution 1, we need to solve two problems simultaneously, namely, the controller design problem that satisfies the robust optimization criteria and the problem of finding controller parameters. The complexity of Solution 1 is high, and the problem of finding parameters can be inexhaustible if the selected controller structure is not suitable. In contrast, Solution 2 usually finds a low-order controller in all cases.

This article designs a low-order robust controller for the self-balancing vehicle control system according to Solution 2. After designing a robust controller, we propose using order reduction algorithms to reduce the order of the controller. The purpose of reducing the order of the controller is to make the control program code simpler and straightforward to practical applications while the system still meets the requirements of robust stability.

The rest of the article is structured as follows. The dynamics and mathematical model of the two-wheel vehicle

are described in Section 2. Section 3 introduces steps to design a robust controller for an uncertain object and then a two-wheel vehicle. Section 4 presents the order reduction of a high-order controller and the simulation and testing of a balanced control system using a low-order controller. Finally, the conclusion is given in Section 5.

2. Dynamics and Mathematical Model of Self-Balancing Two-Wheel Vehicle

2.1. Dynamic of Self-Balancing Two-Wheel Vehicle. The self-balancing two-wheel vehicle, built at the Thai Nguyen University of Technology, is used as the model to verify control algorithms' performance. The vehicle is designed on a children's bicycle, as shown in Figure 1. It is designed to move forward or backward, carry loads, and change loads without falling over.

The self-balancing vehicle is made up of the following components: a flywheel with a diameter of 0.26 m and a weight of 3,796 kg is used to generate torque; DC motor, 15 V, 100 w, and 3400 rpm, is used to rotate the flywheel around its shaft; a DC motor is used to control the vehicle forwards or backward; H-bridge circuit is used to control the DC motor; a 5 VDC 100-pulse Sharp Encoder is used to measure flywheel speed; a sensor GY-521 MPU-6050 is used to measure the tilt angle of the vehicle. The dimensions of the vehicle are as follows: 1.19 m long, 0.5 m high, and 0.4 m wide.

The two-wheel vehicle's balance system is built based on the inverted pendulum principle. When the motor spins the flywheel around its axis with a certain acceleration, a torque is generated, acting on the vehicle to keep the vehicle balanced.

The balance control system configuration of the vehicle is shown in Figure 2. The control program of the self-balancing two-wheel vehicle is loaded into an Adruino Uno to control the balance of the two-wheel vehicle.

2.2. Modeling of Self-Balancing Two-Wheel Vehicle. To build the dynamic model, we make the following assumptions: all vehicle components form a rigid bond with a degree of freedom (DOF) orbiting the Z-axis. The flywheel is firmly attached to the vehicle, and the center of gravity (CoG) is fixed to the chassis. The simplified model of the vehicle is shown in Figure 3, where F is the center of the flywheel and B is the center of the vehicle. The tilt angle of the vehicle around the Z-axis is defined as α , the rotation angle of the flywheel is defined as δ .

The dynamic model of the two-wheel vehicle is shown in Figure 3.

Since the center of the flywheel is not moving relative to the CoG of the vehicle, the absolute speed at points F and B is given as follows:

$$\begin{aligned} |v_B| &= h_b \dot{\alpha}, \\ |v_F| &= h_f \dot{\alpha}, \end{aligned} \quad (1)$$

where h_b is the height of the vehicle's CoG (without the flywheel) and h_f is height of the flywheel's CoG.

Set T_b as the kinetic energy of the two-wheel vehicle and T_f as the kinetic energy of the flywheel. We have T_b and T_f calculated as follows:

$$\begin{aligned} T_b &= \frac{1}{2}m_b|v_B|^2 + \frac{1}{2}I_b\dot{\alpha}^2, \\ T_f &= \frac{1}{2}m_f|v_F|^2 + \frac{1}{2}I_f(\dot{\alpha} + \dot{\delta})^2, \end{aligned} \quad (2)$$

where m_b is the weight of the vehicle including the motors, m_f is the weight of the flywheel, I_b is the moment of inertia of the vehicle, and I_f is the moment of inertia of the flywheel.

The total kinetic energy of the system is given by the following: $T = T_b + T_f$, where

$$T = \frac{1}{2}m_b|v_B|^2 + \frac{1}{2}m_f|v_F|^2 + \frac{1}{2}I_b\dot{\alpha}^2 + \frac{1}{2}I_f\dot{\alpha}^2 + \frac{1}{2}I_f\dot{\delta}^2 + I_f\dot{\alpha}\dot{\delta}, \quad (3)$$

$$T = \frac{1}{2}(m_bh_b^2 + m_fh_f^2 + I_b + I_f)\dot{\alpha}^2 + \frac{1}{2}I_f\dot{\delta}^2 + I_f\dot{\alpha}\dot{\delta}. \quad (4)$$

Defining V as the total energy of the system, we have the following:

$$V = g \cdot \cos \alpha \cdot (m_bh_b + m_fh_f). \quad (5)$$

To describe the dynamics of the two-wheel model, we use the Lagrange equation [8] as follows:

$$\frac{d}{dt} \left[\frac{\partial T}{\partial \dot{q}_i} \right] - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i, \quad (6)$$

where Q_i is the external force and q_i is the general coordinate system.

For $q_i = \alpha$, applying (3)–(6), we have the following:

$$(m_bh_b^2 + m_fh_f^2 + I_b + I_f)\ddot{\alpha} + I_f\ddot{\delta} - g \cdot \sin \alpha \cdot (m_bh_b + m_fh_f) = 0. \quad (7)$$

For $q_i = \delta$, applying (3)–(6), we have the following:

$$I_f\ddot{\alpha} + I_f\ddot{\delta} = C_m, \quad (8)$$

where C_m is the torque of the DC motor shaft attached to the flywheel by a shaft coupling with a gear ratio k:1, determined by the following formula:

$$C_m = kT_m i = kT_m \left[\frac{U - T_e \dot{\delta}}{R} \right], \quad (9)$$

where R is the resistance of the motor; T_m is the torque constant; T_e is the electromotive force constant of the motor.

Applying (9) into (8), we have

$$I_f\ddot{\alpha} + I_f\ddot{\delta} = kT_m \left[\frac{U - T_e \dot{\delta}}{R} \right]. \quad (10)$$

The set of equations for the vehicle kinematic is (7) and (10). We see that these equations are nonlinear.

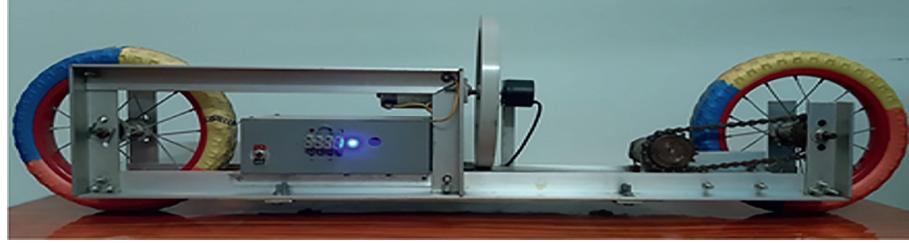


FIGURE 1: The self-balancing two-wheel vehicle.

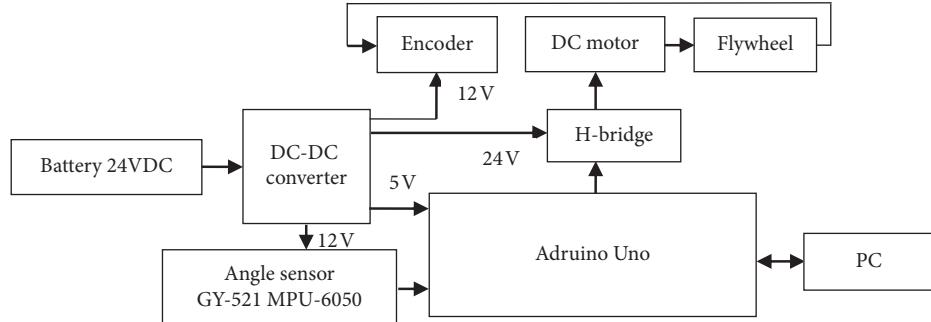


FIGURE 2: The control structure of the self-balancing two-wheel vehicle.

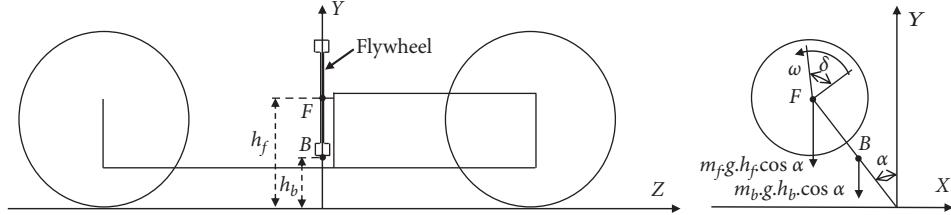


FIGURE 3: Dynamic model of the self-balancing two-wheel vehicle.

By linearizing (7) around the equilibrium point ($\alpha = \delta = 0$, $\sin \alpha = \alpha$) assuming that the tilt angle of the vehicle when operating is tiny ($\alpha \leq 10^0$), we obtain

$$(m_b h_b^2 + m_f h_f^2 + I_b + I_f) \ddot{\alpha} + I_f \ddot{\delta} - g \cdot \alpha \cdot (m_b h_b + m_f h_f) = 0. \quad (11)$$

Set $E = (m_b h_b^2 + m_f h_f^2 + I_b + I_f)$, $F = (m_b h_b + m_f h_f)$.

Define $x = \begin{bmatrix} \alpha = x_1 \\ \dot{\alpha} = x_2 \\ \dot{\delta} = x_3 \end{bmatrix}$ to be the state variable, $y = \alpha$, $u = U$

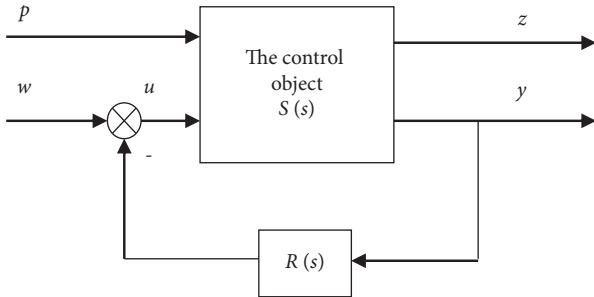
Combining (10) and (11), we obtain a set of equations describing vehicle states as follows:

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y &= Cx + Du, \end{aligned} \quad (12)$$

with the following system parameters:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{Fg}{(E - I_f)} & 0 & \frac{kT_m T_e}{R(E - I_f)} \\ -\frac{Fg}{(E - I_f)} & 0 & -kT_m T_e \frac{E}{I_f R(E - I_f)} \end{bmatrix}, \quad (13)$$

$$B = \begin{bmatrix} 0 \\ \frac{kT_m}{R(E - I_f)} \\ -kR_m \frac{E}{I_f R(E - I_f)} \end{bmatrix}, \quad C = [1 \ 0 \ 0], \quad D = 0.$$

FIGURE 4: Structure diagram of RH_{∞} robust control system.

3. Design of Robust Control for the Self-Balancing Two-Wheel Vehicle

3.1. Design Optimal RH_{∞} for Uncertain Object. Consider a SISO control system according to the structure diagram shown in Figure 4, where $S(s)$ is the model of control object, $R(s)$ is controller, p is the unwanted signal acting on the system, and z is the unwanted output.

The design of RH_{∞} controller consists of two steps:

Step 1. Define the set of all the controllers that make the closed system from w to y stable. This set is denoted by $\tilde{R}(s)$.

Step 2. Find specific element $R(s)$ in set $\tilde{R}(s)$ such that it has the smallest sensitivity to model error and noise and the smallest $\|G_{p \rightarrow z}(s)\|_{\infty}$.

3.1.1. Determine Set $\tilde{R}(s)$ of Controllers to Make the SISO System Stable. (a) Case $S(s)$ is stable. Consider the system containing SISO stages with the structure shown in Figure 4, assuming object $S(s)$ is stable or $S(s) \in RH_{\infty}$. The objective is to determine set $\tilde{R}(s)$, including all controllers, to stabilize the system. Define the following:

$$Q(s) = \frac{R}{1 + RS} \Rightarrow R(s) = \frac{Q}{1 - QS}, \quad (14)$$

With the above controller, the transfer function of the closed system $G_w^y(s)$, i.e., the transfer function between the input signal $w(t)$ and the output signal $y(t)$, is

$$G_w^y(s) = \frac{S}{1 + SR} = (1 - SQ)S. \quad (15)$$

Thus, if function $Q(s)$ is stable that means $Q(s) \in RH_{\infty}$ and $G_w^y(s)$ is also a stable function; i.e., the closed system is robustly stable with the output noise. Accordingly, if the closed system has a stable object $S(s) \in RH_{\infty}$, all controllers $R(s)$ determined by (14) would make the closed system stable and robust with noise. In other words, set $R(s)$, including all the controllers that stabilize the system, has the following Q-dependent form:

$$\tilde{R}(s) = \left\{ R(s) = \frac{Q}{1 - QS} \mid Q \in RH_{\infty} \right\}. \quad (16)$$

(b) Case $S(s)$ is unstable.

When

$$S(s) = \frac{B(s)}{A(s)} = \frac{b_0 + b_1 s + \dots + b_m s^m}{a_0 + a_1 s + \dots + a_n s^n} \text{ with } m \leq n, \quad (17)$$

is unstable, it means that $S(s) \notin RH_{\infty}$ or $A(s)$ is not a Hurwitz polynomial. We can always convert $S(s)$ back to case (a) described above by converting

$$S(s) = \frac{N(s)}{M(s)} \text{ with } N, M \in RH_{\infty}. \quad (18)$$

That is, S will be the real function-rational, with both $N(s)$ and $M(s)$ being stable functions. Indeed, we divide both $B(s)$ and $A(s)$ of (17) by some Hurwitz polynomial $C(s)$:

$$C(s) = c_0 + c_1 s + \dots + c_l s^l, \quad n \leq l. \quad (19)$$

We will get two stable functions:

$$N(s) = \frac{B(s)}{C(s)} \text{ and } M(s) = \frac{A(s)}{C(s)}. \quad (20)$$

Assume the controller $R(s)$ has the following structure:

$$R(s) = \frac{U(s)}{V(s)} \text{ with } U, V \in RH_{\infty}. \quad (21)$$

The transfer function $G_w^y(s)$ of the closed system, describing the relationship between the input signal $w(t)$ and the output signal $y(t)$, is as follows:

$$G_w^y(s) = \frac{S}{1 + SR} = \frac{NV}{NU + MV}. \quad (22)$$

Obviously, if there is a Bezout similarity relation,

$$NU + MV = 1, \quad (23)$$

then $G_w^y(s)$ is a stable function; i.e., the closed system is robustly stable.

Thus, the problem of determining the controller (21) to make the system stable is only finding $U, V \in RH_{\infty}$ to satisfy (23).

However, the Bezout equation (23) has infinite solutions because once there is one solution,

$$NU + MV = 1, \quad (24)$$

then all functions

$$\begin{aligned} U &= X + MQ, \\ V &= Y - NQ, \end{aligned} \quad (25)$$

with all parameters $Q \in RH_{\infty}$ will also be its solution. Therefore, if the object has the following transfer function,

$$S(s) = \frac{N(s)}{M(s)}, \quad (26)$$

with $M, N \in RH_{\infty}$, set $\tilde{R}(s)$ including all controllers that make the closed system stable would have the following structure:

$$\tilde{R}(s) = \left\{ R(s) = \frac{X + MQ}{Y - NQ} \mid X, Y, Q \in RH_{\infty} \text{ và } NX + MY = 1 \right\}. \quad (27)$$

To find $\tilde{R}(s)$, it usually follows the Youla parameterization method [37, 38].

3.1.2. Find $R(s)$ in $\tilde{R}(s)$ So That the System Has the Smallest Sensitivity

- (a) Determine the sensitivity function with respect to the model error.

For a simplicity purpose, we consider the closed system in Figure 4. The closed-loop transfer function of the system is given by the following:

$$G_w^y(s) = \frac{SR}{1 + SR}. \quad (28)$$

Because the object contains deviations ΔS , there is also a corresponding deviation ΔG in the closed system transfer

function. It represents the performance deviation caused by ΔS . The sensitivity function representing the effect of ΔS on system performance is defined as follows:

$$K(s) = \lim_{\Delta S \rightarrow 0} \frac{(\Delta G/G)}{(\Delta S/S)} = \frac{dG}{dS} \cdot \frac{S}{G} = \frac{1}{1 + SR}. \quad (29)$$

The next control objective is to determine $R^*(s) \in R(s)$ that has the potential to minimize the sensitivity, i.e.,

$$\|K(s)\|_\infty = \left\| \frac{1}{1 + S(s)R(s)} \right\|_\infty \longrightarrow \min. \quad (30)$$

- (b) Convert the optimization problem (30) into a model equilibrium problem.

Substituting (18) and (27) into (30), we get the following:

$$\begin{aligned} K(s) &= \frac{1}{1 + R(s)S(s)} = \frac{MY - NQM}{MY - NQM + NX + NMQ} = \frac{MY - NQM}{MY + NX} = T - UQ, \\ T &= YM, \\ U &= NM. \end{aligned} \quad (31)$$

Then, the optimization problem (30) becomes the following:

$$Q^* = \arg \min_{Q \in RH_\infty} \|T - UQ\|_\infty, \quad (32)$$

where $T, U \in RH_\infty$ are known.

- (c) Find the optimal problem solution (32) for the SISO system.

Consider the special case $T, U \in RH_\infty$, where $U(s)$ has a unique zero s_0 , on the right side of the imaginary axis (including $s_0 = \infty$).

According to the definition of the infinity norm, we have the following:

$$\|T - UQ\|_\infty = \sup_{\text{Re}(s) > 0} |T - UQ| \geq |T(s_0) - U(s_0)Q(s_0)| = |T(s_0)|. \quad (33)$$

Besides, for $Q = (T(s) - T(s_0))/U(s)$ (*), we have $|T - UQ| = |T(s_0)|$,

Q in (*) has poles including all poles of T and the remaining zeros (except s_0). That's why Q is a robust function, or $Q \in RH_\infty$.

Therefore, if $(T(s) - T(s_0))/U(s)$ is valid, then $Q = (T(s) - T(s_0))/U(s)$ is the solution of the optimization problem (32). Consequently,

$$R^* = \frac{XU - M(T(s) - T(s_0))}{YU - N(T(s) - T(s_0))}, \quad (34)$$

where:

$$S = \frac{N}{M}, \quad X, Y, Q, N, M \in RH_\infty \text{ and } NX + MY = 1. \quad (35)$$

Expanding the above particular case to the problem with $U(s)$ having m zeros, s_1, s_2, \dots, s_m , on the right side of the imaginary axis, we will apply the Nevanlinna–Pick interpolation formula to transform the problem into a form with only $m - 1$ zeros, s_1, s_2, \dots, s_{m-1} , on the right side of the imaginary axis. Then, we apply Nevanlinna–Pick again and again until the problem has only a unique zero s_1 , on the right of the imaginary axis. In this last case, we will obtain the solution R^* calculated by (34).

3.2. Design of Robust Control for Self-Balancing Two-Wheel Vehicle. Parameters of the self-balancing two-wheel vehicle are given in Table 1. By substitution of these parameters into equation (10), the nominal transfer function of the balancing system of vehicle is described as follows:

$$W(s) = \frac{\alpha(s)}{U(s)} = \frac{-0.223s}{s^3 + 0.1284s^2 - 47.2s - 5.589}, \quad (36)$$

where U is the voltage applied to the DC motor to rotate the flywheel and α is the tilt angle of the vehicle.

The actual operation of self-balancing two-wheel vehicle shows that a change in the vehicle's load will lead to a change in the vehicle's CoG. Besides, external force and uncertain noise due to changes in the vehicle's terrain are uncertain factors affecting the vehicle. These uncertainties alter the self-balancing two-wheel vehicle model, so we can consider the two-wheel vehicle model as an uncertain object. Among the uncertainties affecting the vehicle, we mainly concern about the load variation. Specifically, consider the following cases.

TABLE 1: Parameters of the self-balancing two-wheel vehicle.

Parameters	Value	Unit
I_b	0,1105	Kg.m ²
h_b	0,105	m
I_f	0,03289	Kg.m ²
h_f	0,205	m
m_b	10,024	Kg
m_f	3,976	Kg
T_e	0,045	V.s/Rad
T_m	0,045	Nm/A
R	0,52	Ω
k	1 : 1	
g	9,81	m/s ²

Case 1. When load is 1.8 kg, the height of the CoG of the vehicle $h_b = 0.13m$ is the transfer function of the vehicle model as follows:

$$W_1(s) = \frac{\alpha(s)}{U(s)} = \frac{-0.1527s}{s^3 + 0.1253s^2 - 40.72s - 4.821}. \quad (37)$$

Case 2. When the load is 2.9 kg, the height of the CoG of the vehicle $h_b = 0.16m$ is the transfer function of the vehicle model as follows:

$$W_2(s) = \frac{\alpha(s)}{U(s)} = \frac{-0.3735s}{s^3 + 0.1352s^2 - 61.37s - 7.352}. \quad (38)$$

Case 3. When the load is 5 kg, the height of the CoG of the vehicle $h_b = 0.05m$ is the transfer function of the vehicle model as follows:

$$\begin{aligned} A(s) = & -1.01s^{15} - 37.98s^{14} - 646s^{13} - 6580s^{12} - 4.486 \cdot 10^4 s^{11} - 2.169 \cdot 10^5 s^{10} - 7.7 \cdot 10^5 s^9 - 2.051 \cdot 10^6 s^8 \\ & - 4.147 \cdot 10^6 s^7 - 6.393 \cdot 10^6 s^6 - 7.46 \cdot 10^6 s^5 - 6.453 \cdot 10^6 s^4 - 3.951 \cdot 10^6 s^3 - 1.566 \cdot 10^6 s^2 - 3.338 \cdot 10^5 s \\ & - 2.302 \cdot 10^4, \end{aligned} \quad (42)$$

$$\begin{aligned} B(s) = & 2.23 \cdot 10^{-5} s^{15} + 0.00525 s^{14} + 0.1697 s^{13} + 2.401 s^{12} + 18.97 s^{11} + 92.73 s^{10} + 292.1 s^9 + 589.7 s^8 \\ & + 707.1 s^7 + 346.7 s^6 - 241 s^5 - 436.9 s^4 - 203 s^3 - 18.37 s^2. \end{aligned}$$

Applying Matlab/Simulink to simulate the two-wheeled vehicle balance control system using the 15th-order controller, we obtained the results shown in Figures 7 and 8.

When the two-wheel vehicle loads the different circumstances, we obtained the results shown in Figure 9 as follows.

(*) Comment: Figures 8 and 9 show that the 15th-order robust controller is capable of maintaining a stable balance of the two-wheel vehicle when the model parameters change. However, using a 15th-order controller to control a two-wheel vehicle results in a complex control program code, which can affect the actual control system's response speed and hence make the system unstable. Therefore, we seek to reduce the order of the 15th-order controller to simplify the

$$W_3(s) = \frac{\alpha(s)}{U(s)} = \frac{-0.3573s}{s^3 + 0.1345s^2 - 63.44s - 7.511}. \quad (39)$$

Case 4. When the load is 10 kg, the height of the CoG of the vehicle $h_b = 0.12m$ is the transfer function of the vehicle model as follows:

$$W_4(s) = \frac{\alpha(s)}{U(s)} = \frac{-0.1774s}{s^3 + 0.1264s^2 - 64.72s - 7.663}. \quad (40)$$

The Bode plots of two-wheel vehicle models with different loads are shown in Figure 5.

As shown in Figure 5, uncertainties can reduce the accuracy of the mathematical model, leading to a decrease in the control performance and even making the system unstable. Due to uncertainty in the two-wheel vehicle model, robust control [11, 34] is the most suitable control algorithm to control the balance of the two-wheel vehicle model among proposed control algorithms, including PD [20], PID [17–19], LQG and MPC [21], and SMC [25–27].

The robust control of a two-wheel vehicle is designed according to the structure diagram given in Figure 6.

Designing the robust controller for the self-balancing two-wheel vehicle according to the procedure presented in Section 3 [35, 36], we obtain the following controller:

$$R(s) = \frac{A(s)}{B(s)}, \quad (41)$$

where

control program code and ensure that the real-time control system always maintains a stable balance.

4. Results and Discussion

4.1. Reducing Order of Controller. The requirement for the order reduction problem is that a low-order controller has the lowest possible order; the response of a control system using a low-order controller should match that of a control system using the 15th-order controller or the control system using a low-order controller ensures stability control for two-wheel vehicle.

To reduce order of the controller, there are two basic solutions as follows:

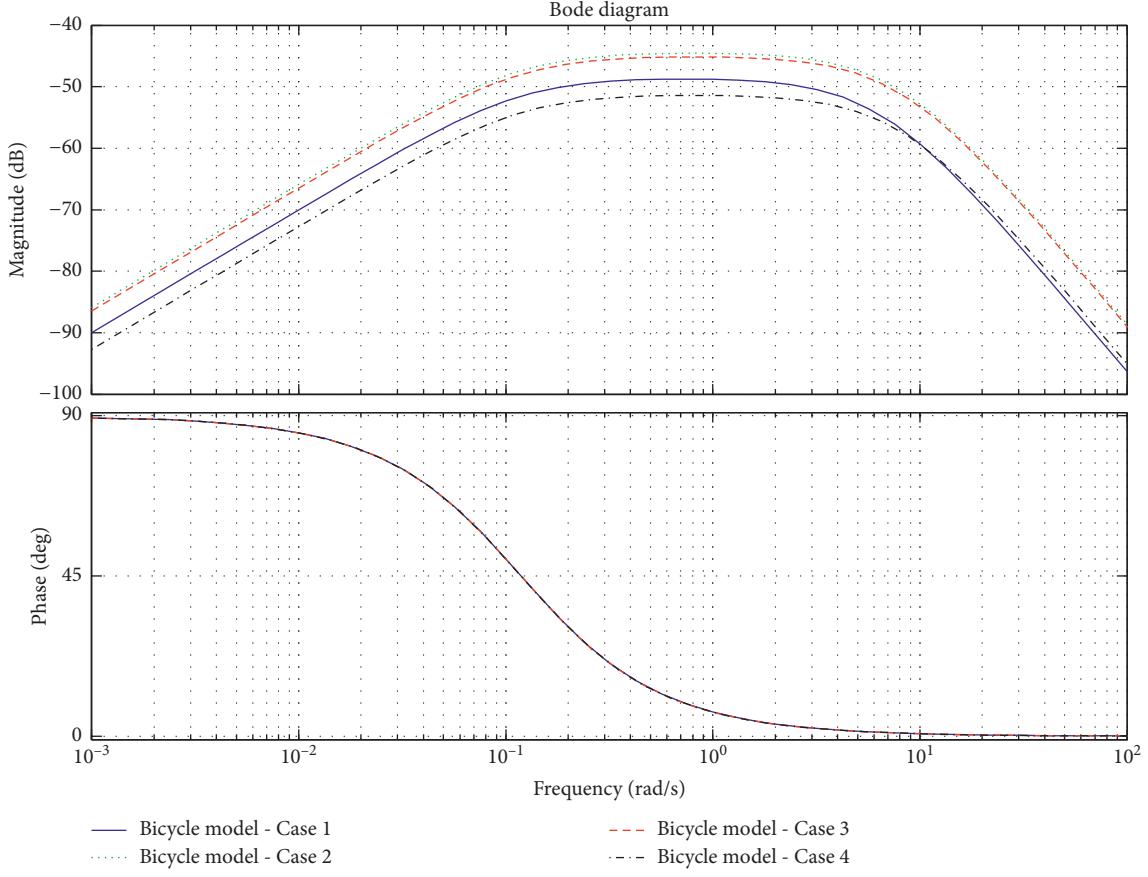


FIGURE 5: Bode plot of the self-balancing two-wheel vehicle mode with different loads.

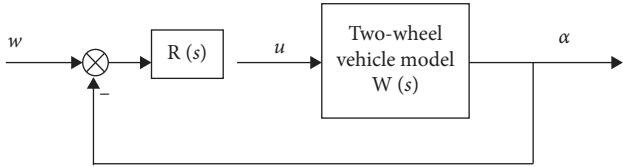


FIGURE 6: Structure of the robust control system for a two-wheel vehicle.

Solution 1: Select a low-order controller with a fixed structure and then apply optimal algorithms to select the parameters of the low-order controller so that the control system using the low-order controller ensures stability control for a two-wheel vehicle.

Solution 2: Applying the order reduction algorithms to reduce the order of a high-order controller, we will obtain a set of reduced-order controllers. The lowest reduced-order controller that still meets the requirements for stable balance control of a two-wheel vehicle will be selected to replace the high-order controller.

In this article, we choose Solution 2 to reduce the order of the 15th-order controller. Analyzing the 15th-order controller, we find that it is an unstable system. In order to reduce the order of an unstable system, we need to use algorithms that reduce the order of an unstable system.

The order reduction algorithms for unstable systems are classified into two groups. One group consists of algorithms for the indirect order reduction of an unstable system. These algorithms divide the unstable system into an unstable subsystem and a stable subsystem. The order of the stable subsystem will be reduced to obtain a low-order stable subsystem. The unstable subsystem will then be added to the low-order stable subsystem to get a low-order unstable system. The other group consists of algorithms for direct order reduction of an unstable system. These algorithms perform order reduction of an unstable system without performing subsystem division. Aiming to select the best low-order controller to replace the 15th-order controller, we use both groups of reduction methods and then compare the obtained results.

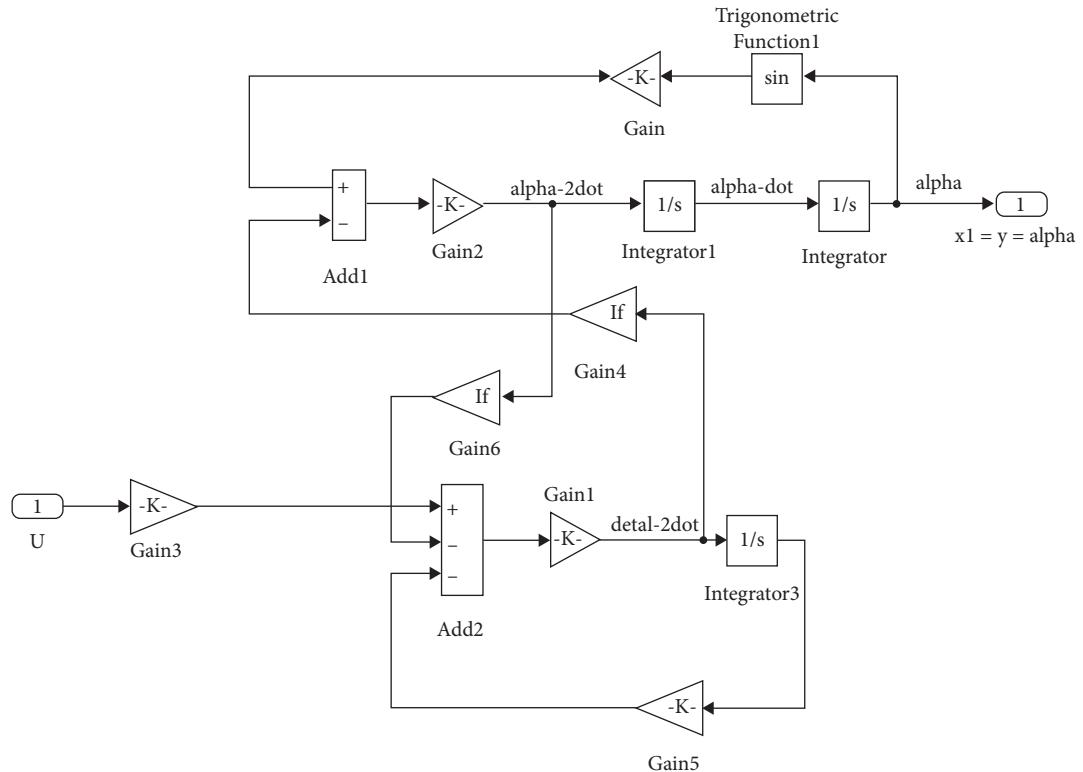
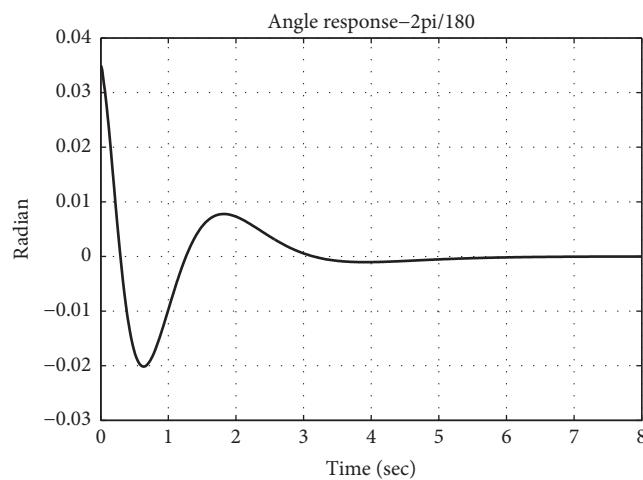


FIGURE 7: Two-wheel vehicle model in Matlab/Simulink.

FIGURE 8: Tilt angle response of the control system with initial angle $2(\pi/180)$ (ra d).

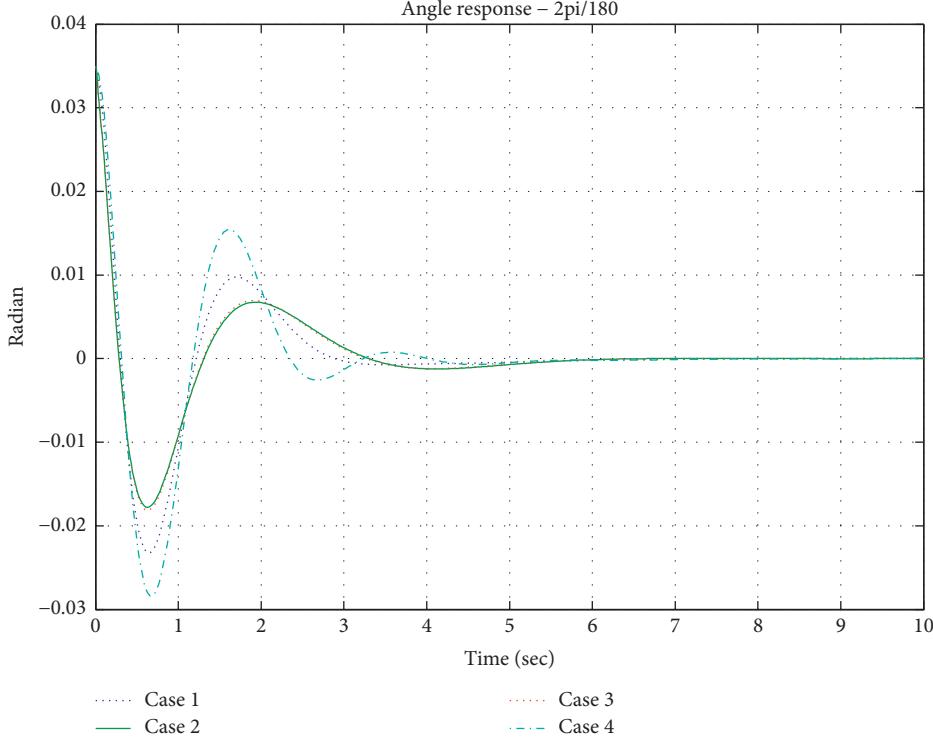


FIGURE 9: Tilt angle response of the control system when the two-wheel vehicle loads the different circumstances with the initially tilt angle $2(\pi/180)(rad)$.

4.1.1. Indirect Order Reduction Algorithms. Separating the 15th-order controller into stable subsystems and unstable

subsystems according to the algorithm [38], we obtained the following results:

The stable subsystem:

$$\mathbf{R}_s(s) = \frac{\mathbf{K}(s)}{\mathbf{M}(s)},$$

$$\begin{aligned} \mathbf{K}(s) = & -4.531 \cdot 10^4 s^{12} - 1.737 \cdot 10^6 s^{11} - 2.972 \cdot 10^7 s^{10} - 2.974 \cdot 10^8 s^9 - 1.928 \cdot 10^9 s^8 - 8.5 \cdot 10^9 s^7 \\ & - 2.61 \cdot 10^{10} s^6 - 5.603 \cdot 10^{10} s^5 - 8.297 \cdot 10^{10} s^4 - 8.149 \cdot 10^{10} s^3 - 4.887 \cdot 10^{10} s^2 \\ & - 1.481 \cdot 10^{10} s - 1.188 \cdot 10^9, \end{aligned} \quad (43)$$

$$\begin{aligned} \mathbf{M}(s) = & s^{12} + 236.3 s^{11} + 7798 s^{10} + 1.138 \cdot 10^5 s^9 + 9.409 \cdot 10^5 s^8 + 4.903 \cdot 10^6 s^7 + 1.698 \cdot 10^7 s^6 \\ & + 3.987 \cdot 10^7 s^5 + 6.324 \cdot 10^7 s^4 + 6.555 \cdot 10^7 s^3 + 4.102 \cdot 10^7 s^2 + 1.284 \cdot 10^7 s + 1.042 \cdot 10^6. \end{aligned}$$

The unstable subsystem:

$$\mathbf{R}_{us}(s) = \frac{-2653 s^2 - 2162 s - 991}{s^3 - 0.7907 s^2 + 1.776 \cdot 10^{-16} s - 6.541 \cdot 10^{-18}}. \quad (44)$$

Since the unstable subsystem does not change during order reduction, the 4th-order controller is the smallest order controller obtained by indirect order reduction algorithms.

Applying some indirect order reduction algorithms [39–47] to reduce the order of the 15th-order controller

(using balancmr/bstmr functions in Matlab/Simulink), we obtained the results as given in Tables 2 and 3.

4.1.2. Direct Order Reduction Algorithms. Applying direct order reduction algorithms [40, 48–50] (using programmed code and the available ncfmr function in Matlab/Simulink), to reduce the order of the 15th-order controller, we obtained the results in Table 4.

TABLE 2: The 4th-order controllers using indirect order reduction algorithms.

Order reduction algorithm	Transfer function of reduced order controller, Rr(s)
Balanced truncation model reduction (balancmr) [39–43]	$(-4.531 \cdot 10^4 s^4 - 8.686 \cdot 10^4 s^3 - 4.417 \cdot 10^5 s^2 - 4.32 \cdot 10^5 s - 1.986 \cdot 10^5 / s^4 + 199.9 s^3 - 158.7 s^2 - 2.782 \cdot 10^{-14} s - 2.132 \cdot 10^{-17})$
Balanced stochastic truncation model reduction (bstmr) [44–47]	$(-4.531 \cdot 10^4 s^4 - 7.535 \cdot 10^4 s^3 - 4.479 \cdot 10^5 s^2 - 4.342 \cdot 10^5 s - 1.985 \cdot 10^5 / s^4 + 199.5 s^3 - 158.4 s^2 - 3.556 \cdot 10^{-14} s - 1.31 \cdot 10^{-15})$

4.2. Simulation Results. To evaluate the efficiency of low-order controllers, we build a simulation model of the self-balancing two-wheel vehicle control system with low-order controllers using Matlab/Simulink; results are obtained as follows.

Using the 4th-order controller as given in Table 2 to control a self-balancing two-wheel vehicle, as shown in Figure 10, we obtain results as depicted in Figure 11.

Figure 11 shows that a self-balancing two-wheeled vehicle control system using the 4th-order controller can not stabilize the vehicle. Thus, the 4th-order controller does not meet the requirements of reducing the order of the 15th-order controller.

Using the 5th-order controllers in Table 3 to control the self-balancing two-wheeled vehicle, as in Figure 12, we obtained results as depicted in Figure 13.

Figure 13 shows that the self-balancing two-wheel vehicle using the 5th-order controller can stabilize the vehicle with an initially off-angle of $(2\pi/180)$ (radian). Comparing the control system performance using the 5th-order controller to the system using the 15th-order controller, we see that the control system using the 5th-order controller, according to the bstmr algorithm, has the largest number of fluctuations and the longest settling time. When the control system using the 5th-order controller according to the balancmr algorithm, the amplitude of the oscillation is small. Meanwhile, the settling time of the control system using the 5th-order controller according to the balancmr algorithm is equivalent to that of the control system using the 15th-order controller.

Using the 2nd-order controllers in Table 3 to control the self-balancing two-wheel vehicle, as shown in Figure 14, we obtained the following results.

Figure 15 shows that the control system using the 2nd-order controller, according to Zilochian's algorithm, can not stabilize the vehicle. In contrast, Figure 16 depicts that the control system using the 2nd-order controller, either according to the ncfmr algorithm or Zhou's algorithm, ensures stability balance control for the self-balancing two-wheel vehicle. Out of these two algorithms, the quality of the control system using the 2nd-order controller, according to

Zhou's algorithm, is better (smaller number of the oscillation and smaller amplitude of the oscillation) than the quality of the control system using the 2nd-order controller according to the ncfmr algorithm.

By comparing and evaluating the reduced-order controllers by direct and indirect order reduction algorithms, we selected the 2nd-order controller according to Zhou's algorithm to control the self-balancing two-wheel vehicle.

Verifying the control system using the 2nd-order controller according to Zhou's algorithm in variable load cases, we obtained the results as shown in Figure 17.

As shown in Figure 17, the self-balancing two-wheel vehicle control system using the 2nd-order controller, according to Zhou's algorithm, can stabilize the vehicle when the vehicle's load changes.

4.3. Experimental Results. The structure of the control system is shown in Figure 2. First, we perform the controller with the model built on Matlab/Simulink. After verifying the control system's control capability using the Simulink model, we loaded the control program using the 2nd-order controller into the Arduino Uno board's chip with a sampling time of 10 ms. The tilt angle of the vehicle will be read by sensor GY-521 MPU-6050. Based on the received tilt angle signal, the control program in the Arduino will calculate the control signal sent to the H-bridge circuit. Accordingly, the H-bridge circuit will generate the appropriate voltage signal to control the DC motor. The DC motor will rotate the flywheel so that the torque induces an impact on the vehicle against the gravity torque due to the vehicle tilting away from its balanced position.

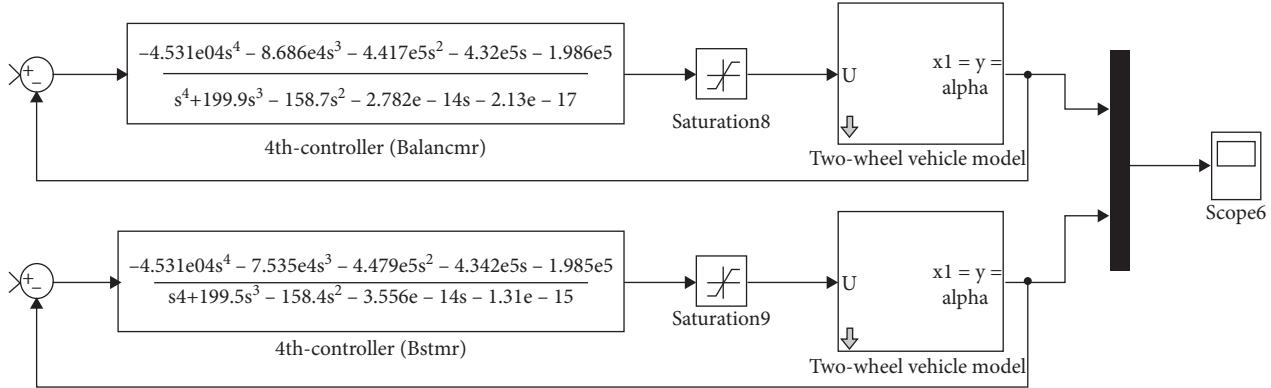
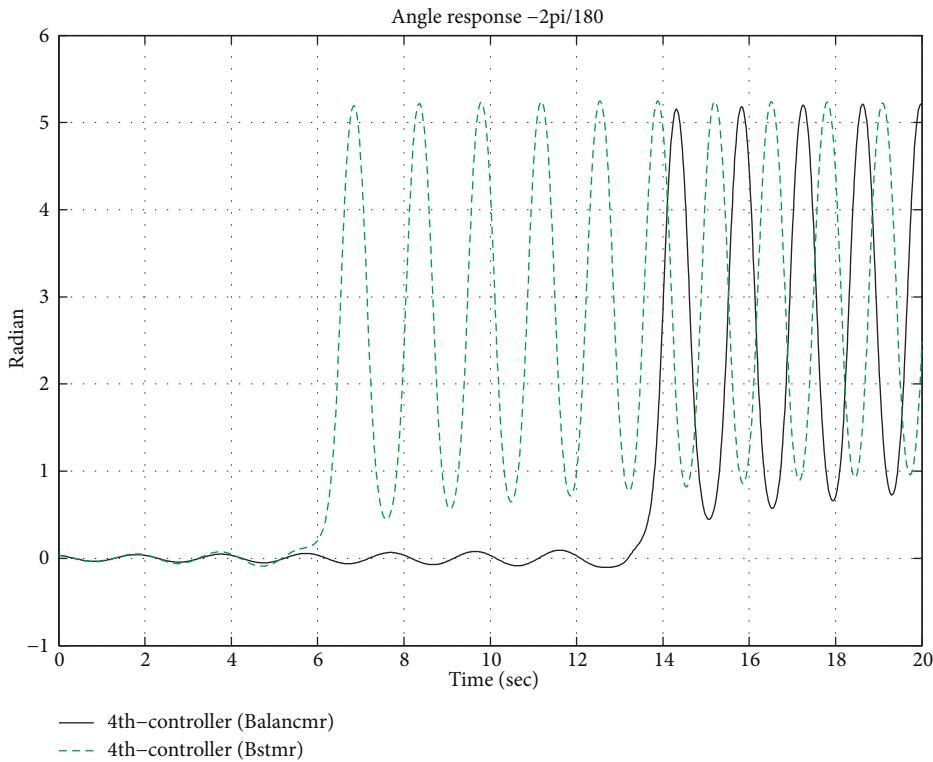
For experimental validation, Arduino is connected to the ESP32 wifi module to measure the vehicle's tilt angle. This measured tilt angle will be sent to the server of the Blynk application by Arduino via the ESP 32 wifi module. Then, the computer will receive the tilt angle information from the server of the Blynk application. Using Matlab/Simulink software installed on the computer, the vehicle's tilt angle is obtained, as shown in the following figures.

TABLE 3: The 5th-order controllers using indirect order reduction algorithms.

Order reduction algorithm	Transfer function of reduced order controller, $R_r(s)$
Balanced truncation model reduction (balancmr) [39–43]	$(-4.531.10^4 s^5 - 4.544.10^5 s^4 - 2.552.10^6 s^3 - 3.14.10^6 s^2 - 3.894.10^6 s - 1.699.10^6/s^5 + 207.9s^4 + 1552s^3 - 1357s^2 - 2.38.10^{-13}s - 1.824.10^{-16})$
Balanced stochastic truncation model reduction (bstmr) [44–47]	$(-4.531.10^4 s^5 - 5.889.10^5 s^4 - 2.731.10^6 s^3 - 4.545.10^6 s^2 - 5.253.10^6 s - 2.311.10^6/s^5 + 210.9s^4 + 2165s^3 - 1844s^2 + 4.128.10^{-13}s - 1.526.10^{-14})$

TABLE 4: The 2nd-order controllers using direct order reduction algorithms.

Order reduction algorithm	Transfer function of reduced order controller – Rr (s)
Zhou's balanced truncation algorithm [48]	$(-4.531 \cdot 10^4 s^2 - 3.001 \cdot 10^5 s - 2.484 \cdot 10^6 s^2 + 204.5s + 898.2)$
Balanced truncation model reduction for normalized coprime factors (ncfmr) [40, 49]	$(-4.531 \cdot 10^4 s^2 - 4.953 \cdot 10^4 s - 2.135 \cdot 10^5 s^2 + 69.66s - 159.4)$
Zilochian's balanced truncation algorithm [50]	$(-4.531 \cdot 10^4 s^2 - 6.714 \cdot 10^4 s - 1.268 \cdot 10^6 s^2 + 199.3s - 158.3)$

FIGURE 10: Simulation diagram of the self-balancing two-wheel vehicle control system using 4th-order controllers.FIGURE 11: Step response of the self-balancing two-wheel vehicle control system using 4th-order controllers.

Experimental control results of self-balancing two-wheeled vehicle when the vehicle is stationary and at no-load are shown in Figure 18 as follows

Figure 18 depicts that the control system can stabilize the vehicle at no-load with the tilt angle fluctuating around 0 degrees, from -0.08 degrees to 0.08 degrees.

The results of experimenting with the vehicle balance control system when the vehicle is stationary and carrying a load of 1.8 kg are obtained, as shown in Figure 19.

Figure 19 shows that the vehicle balance control system can stabilize the vehicle when the vehicle carries a load of 1.8 kg with the tilt angle of the vehicle fluctuating around 0°,

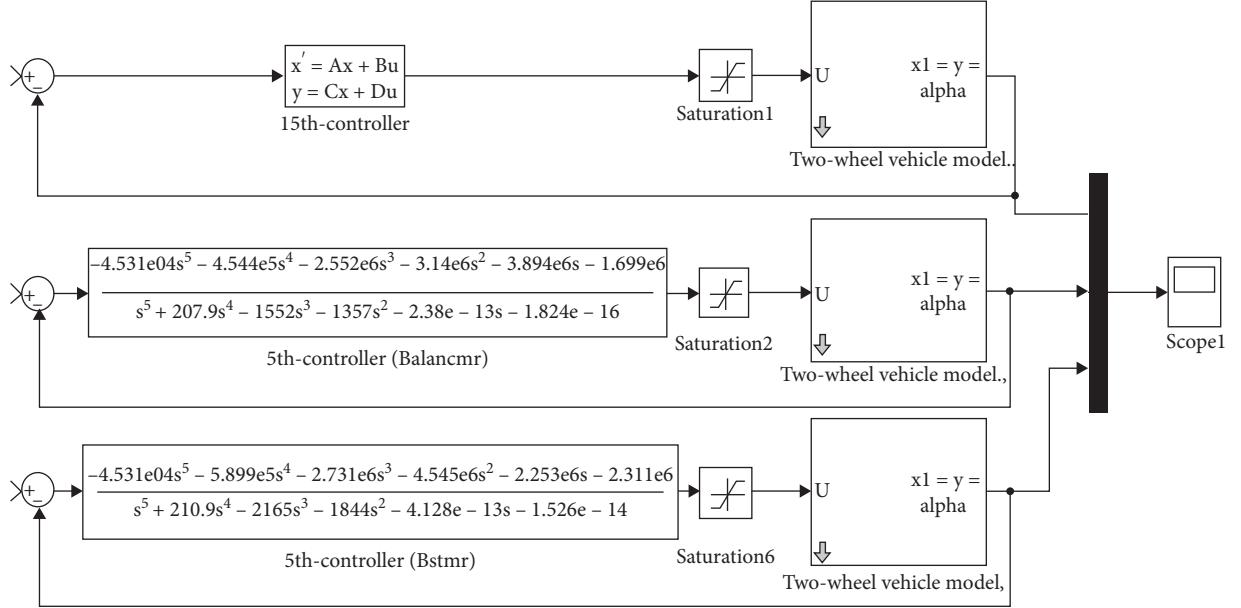


FIGURE 12: Simulation diagram of the self-balancing two-wheel vehicle control system using 5th-order controllers.

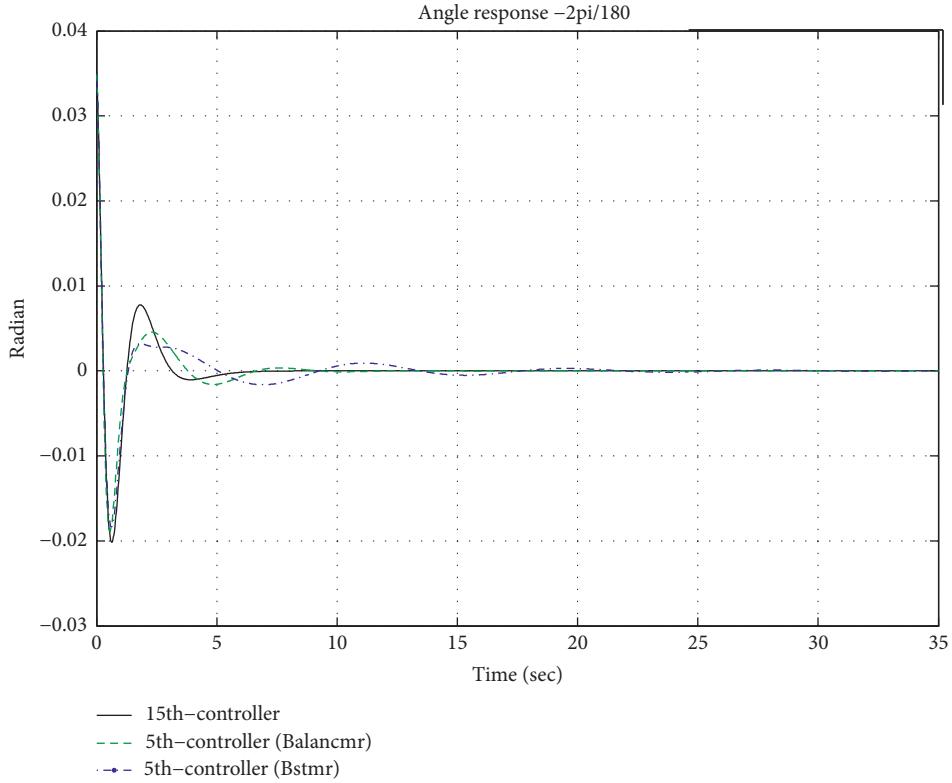


FIGURE 13: Step response of the self-balancing two-wheel vehicle control system using 5th-order controllers.

from -0.2 degrees to 0.2 degrees. Compared to the no-load case, the amplitude of the tilt angle variation of the loaded case is higher.

Experimenting with the vehicle balance control system when the vehicle is stationary and carrying a 2.9 kg load, we obtained the results shown in Figure 20.

Figure 20 shows that the control system can stabilize the vehicle when the vehicle carries 2.9 kg with the tilt angle oscillating around 0 degrees, from -0.6 degrees to 0.6 degrees. Comparing the results between Figures 19 and 20, we see that the angular oscillation amplitude increases when the load weight increases.

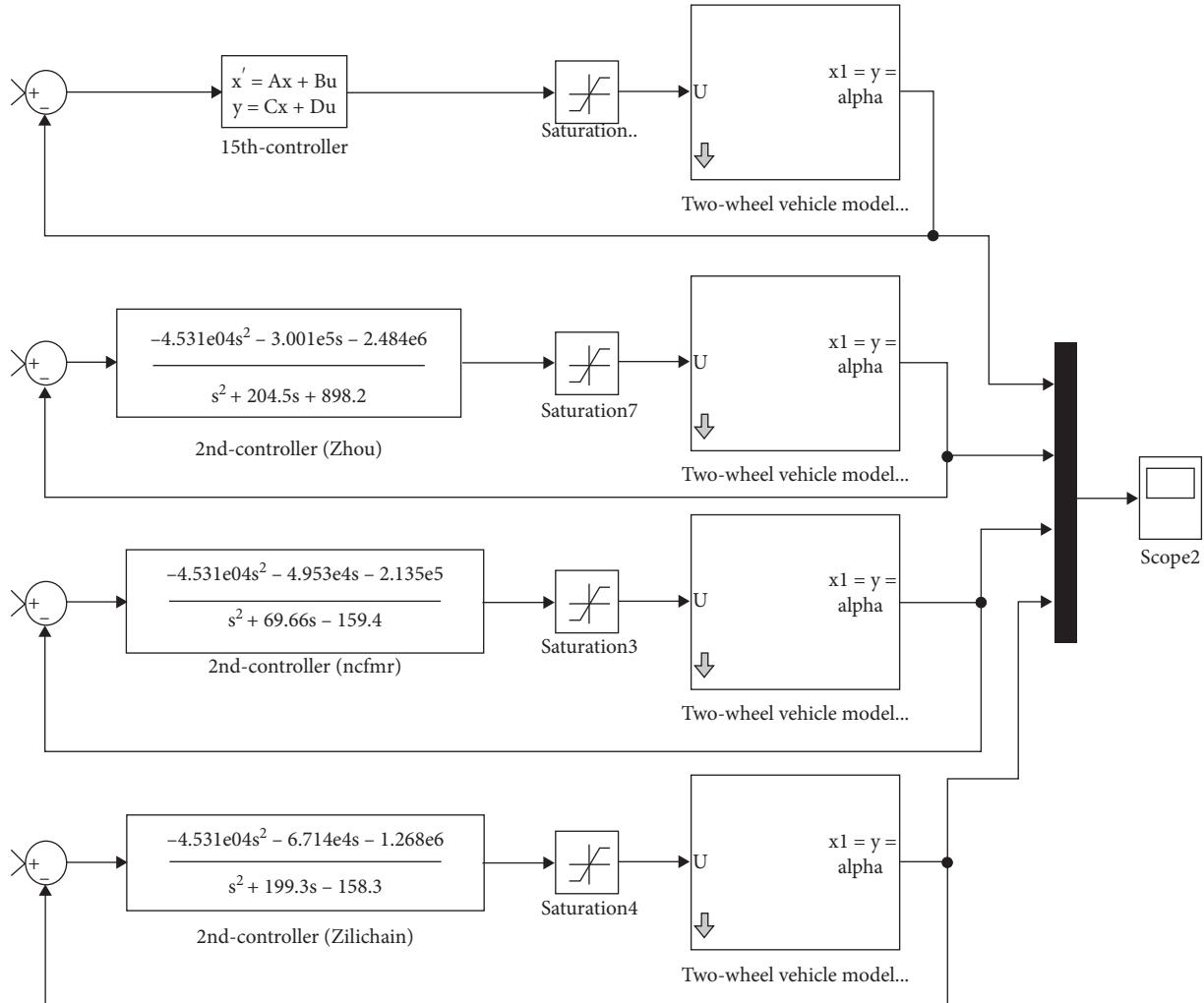


FIGURE 14: Simulation diagram of the self-balancing two-wheel vehicle control system using 2nd-order controllers.

Experimenting with the control system when the vehicle is stationary and placing a load of 1.8 kg at the eccentric position of the vehicle, we obtained the results given in Figure 21.

Figure 21 shows that when the vehicle is not loaded, the control system can stabilize the vehicle with the vehicle's tilt angle fluctuating around 0 degrees. When the vehicle carries a load of 1.8 kg at the eccentric position of the vehicle, the control system automatically finds a new equilibrium with the tilt angle fluctuating around 2 degrees.

Experimenting with the vehicle balance control system when the vehicle moves while carrying a load of 1.8 kg, we obtained the results given in Figure 22.

Figure 22 shows that when the vehicle is not moving, the vehicle's tilt angle fluctuates around 0 degrees with amplitude ranging from -0.2 to 0.2 degrees. When the vehicle is moving, from 5 s to 30 s, the tilt angle will fluctuate around 0 degrees with an amplitude of 0.8 to 1 degree. When the vehicle stops, the tilt angle is stabilized around 0 degrees with the same oscillation amplitude as when the vehicle is not

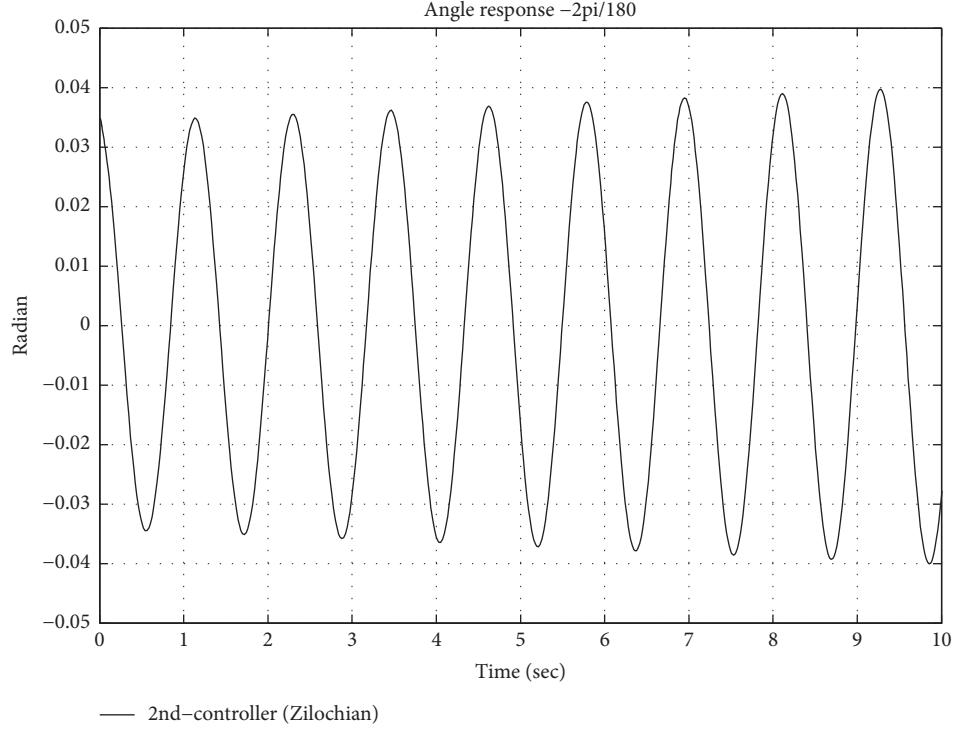


FIGURE 15: Step response of the self-balancing two-wheel vehicle control system using 2nd-order controllers according to Zilochian's algorithm.

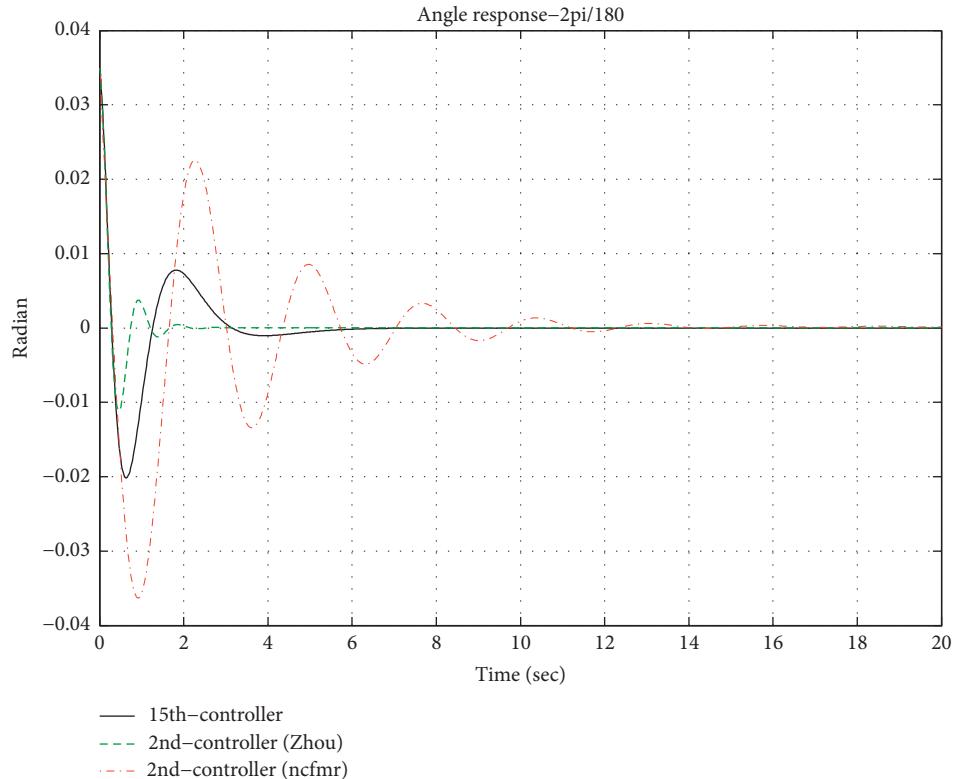


FIGURE 16: Step response of the self-balancing two-wheel vehicle control system using 2nd-order controllers according to Zhou's algorithm and ncfmr algorithm.

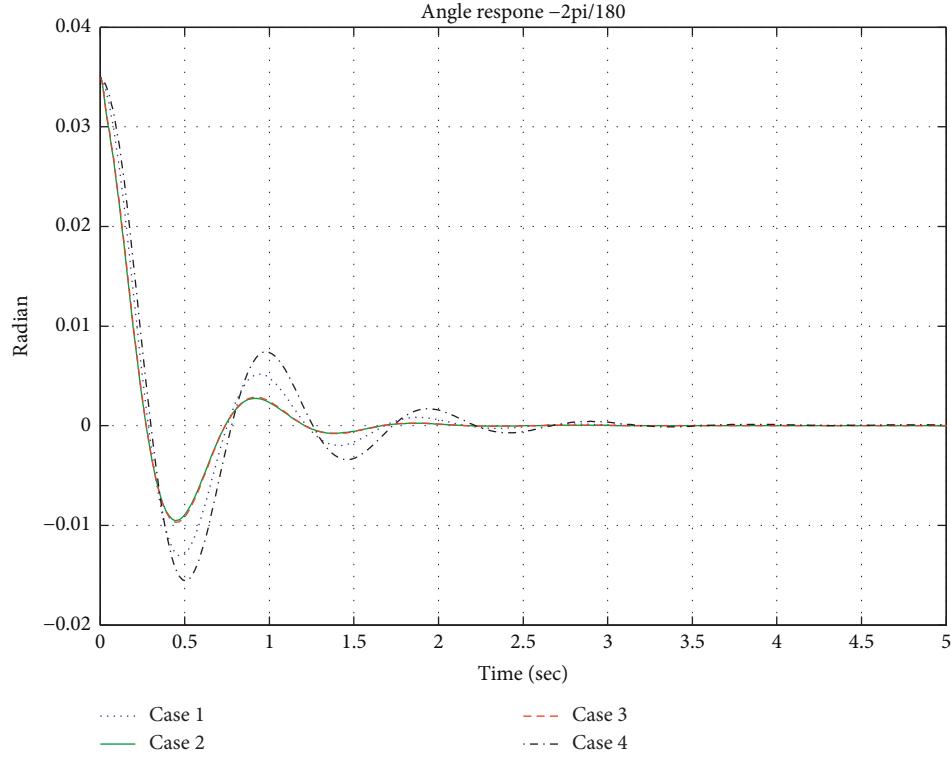


FIGURE 17: Step response of the self-balancing two-wheel vehicle control system using 2nd-order controllers according to Zhou's algorithm when the vehicle's load changes.

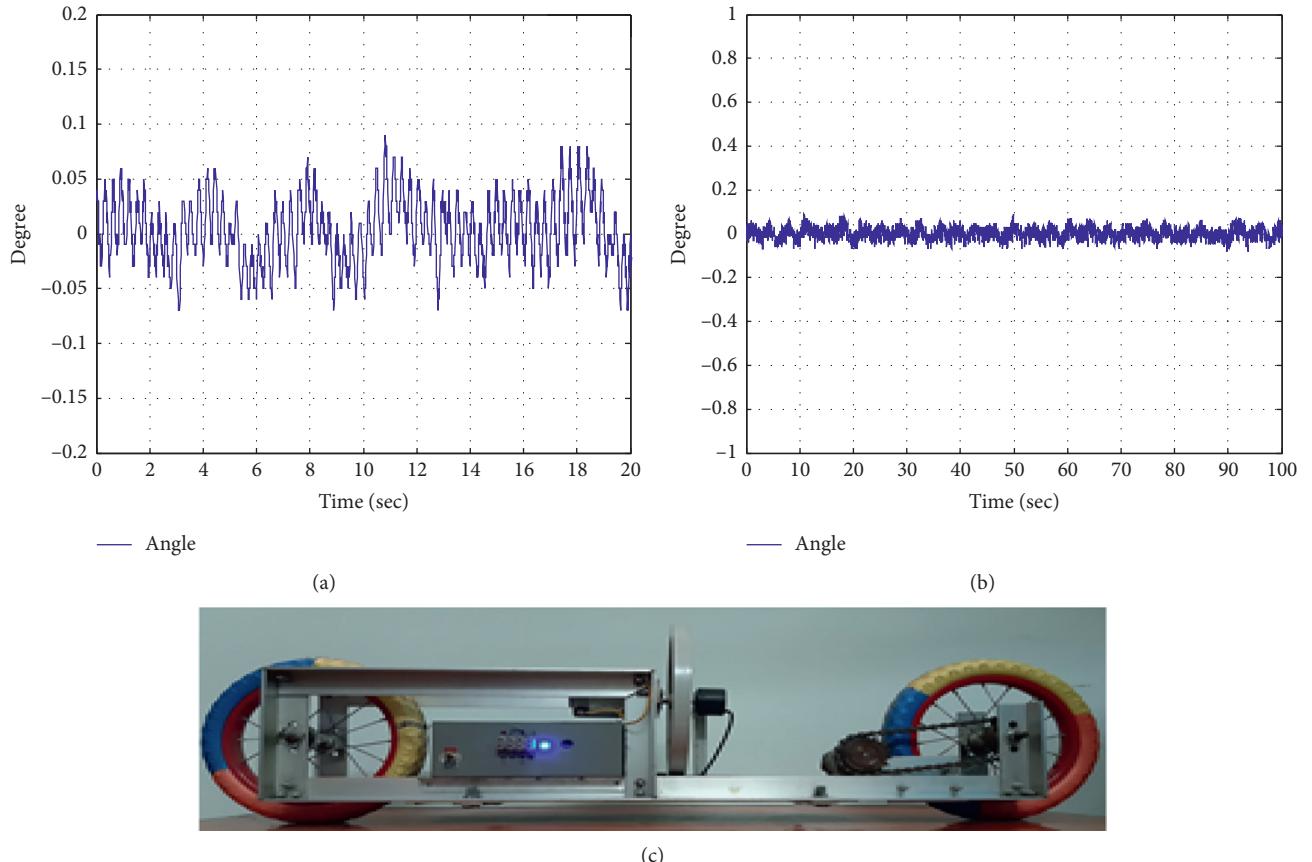


FIGURE 18: Experimental results when the vehicle is stationary: (a) tilt angle response (0–20 seconds); (b) tilt angle response (0–100 seconds); (c) balanced vehicle.

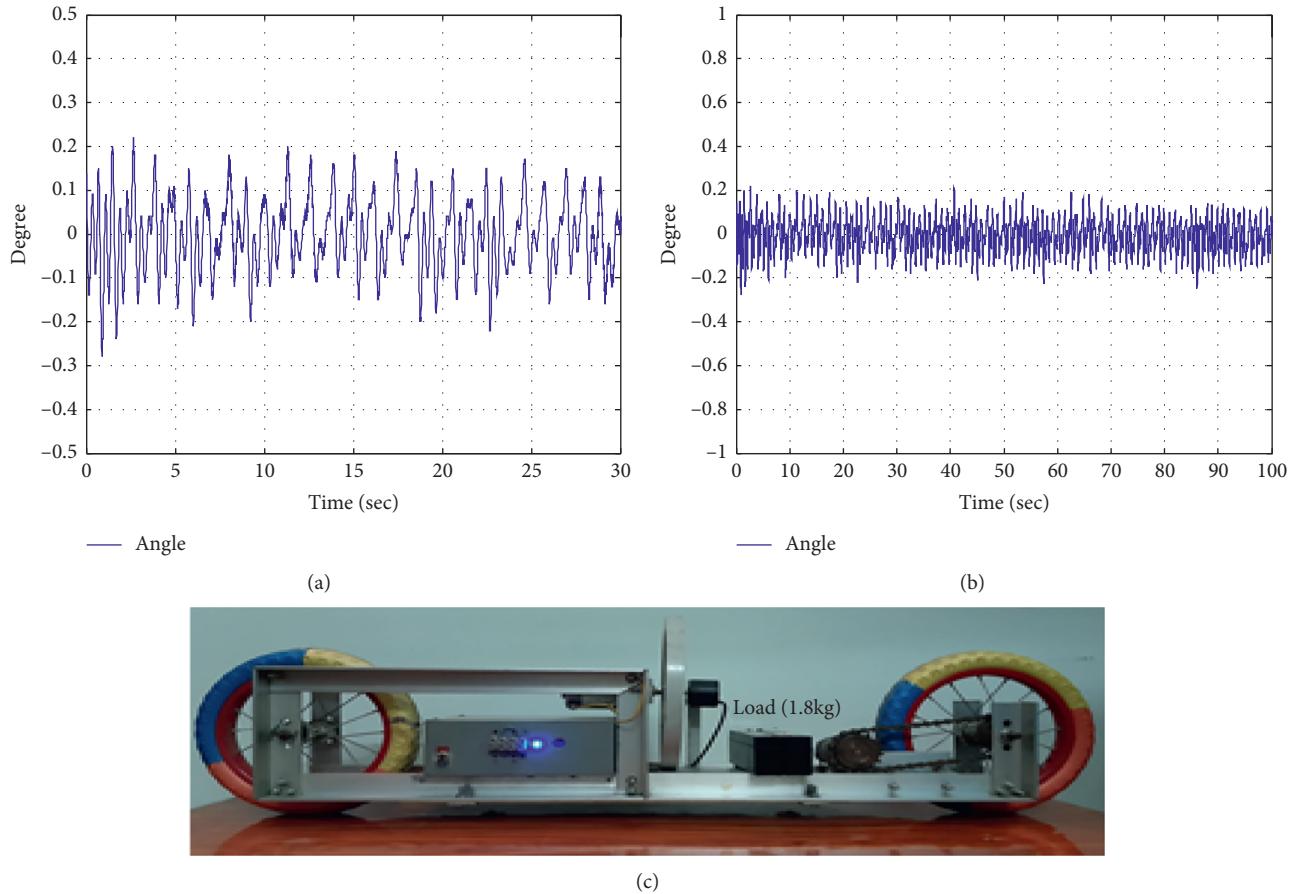


FIGURE 19: Experimental results when the vehicle is stationary and carrying a load of 1.8 kg: (a) tilt angle response (0–30 seconds); (b) tilt angle response (0–100 seconds); (c) balanced vehicle.

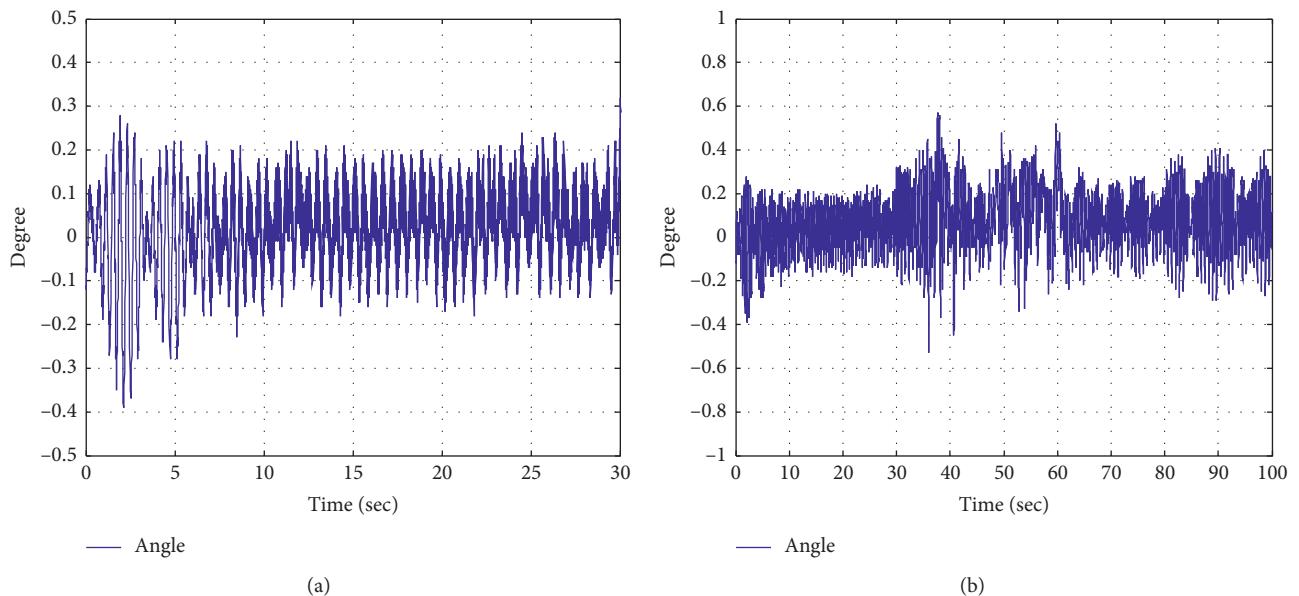


FIGURE 20: Continued.

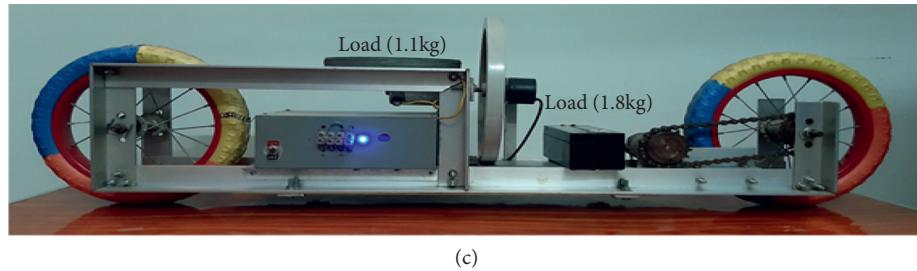


FIGURE 20: Experimental results when the vehicle is stationary and carrying a load of 2.9 kg: (a) tilt angle response (0–30 seconds); (b) tilt angle response (0–100 seconds); (c) balanced vehicle.

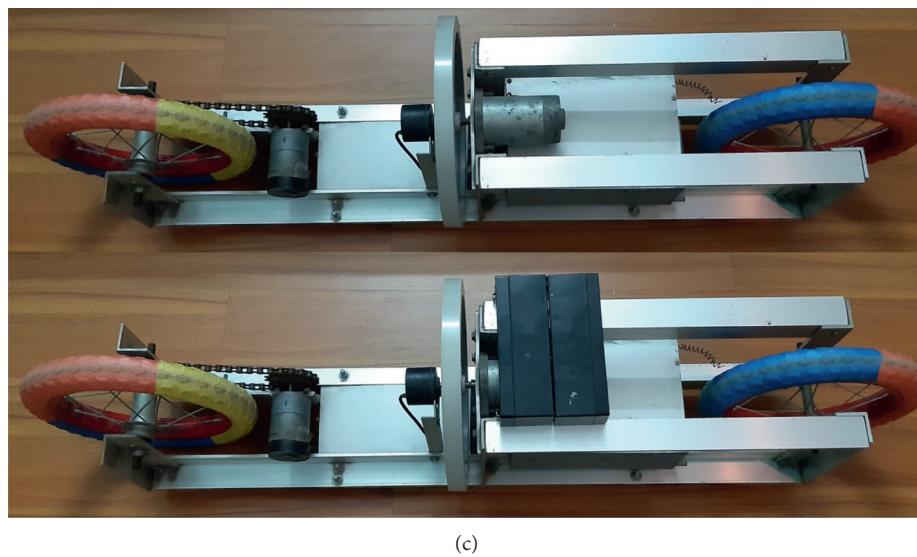
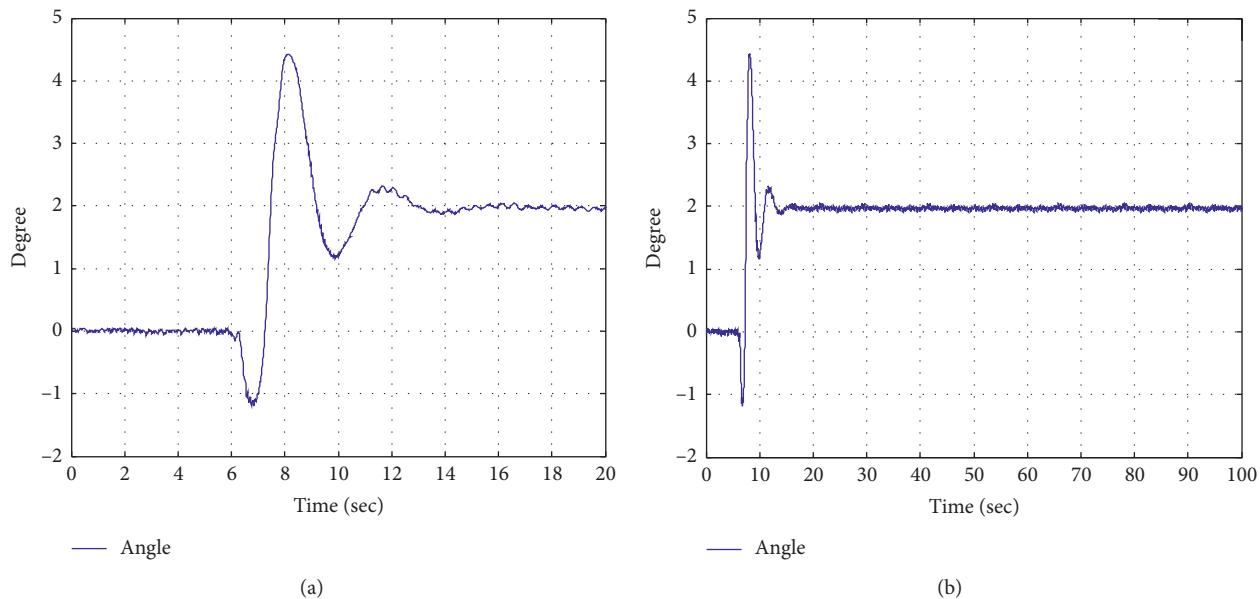


FIGURE 21: Experimental results when the vehicle is stationary and carrying a load of 1.8 kg eccentric: (a) tilt angle response (0–20 seconds); (b) tilt angle response (0–100 seconds); (c) balanced vehicle.

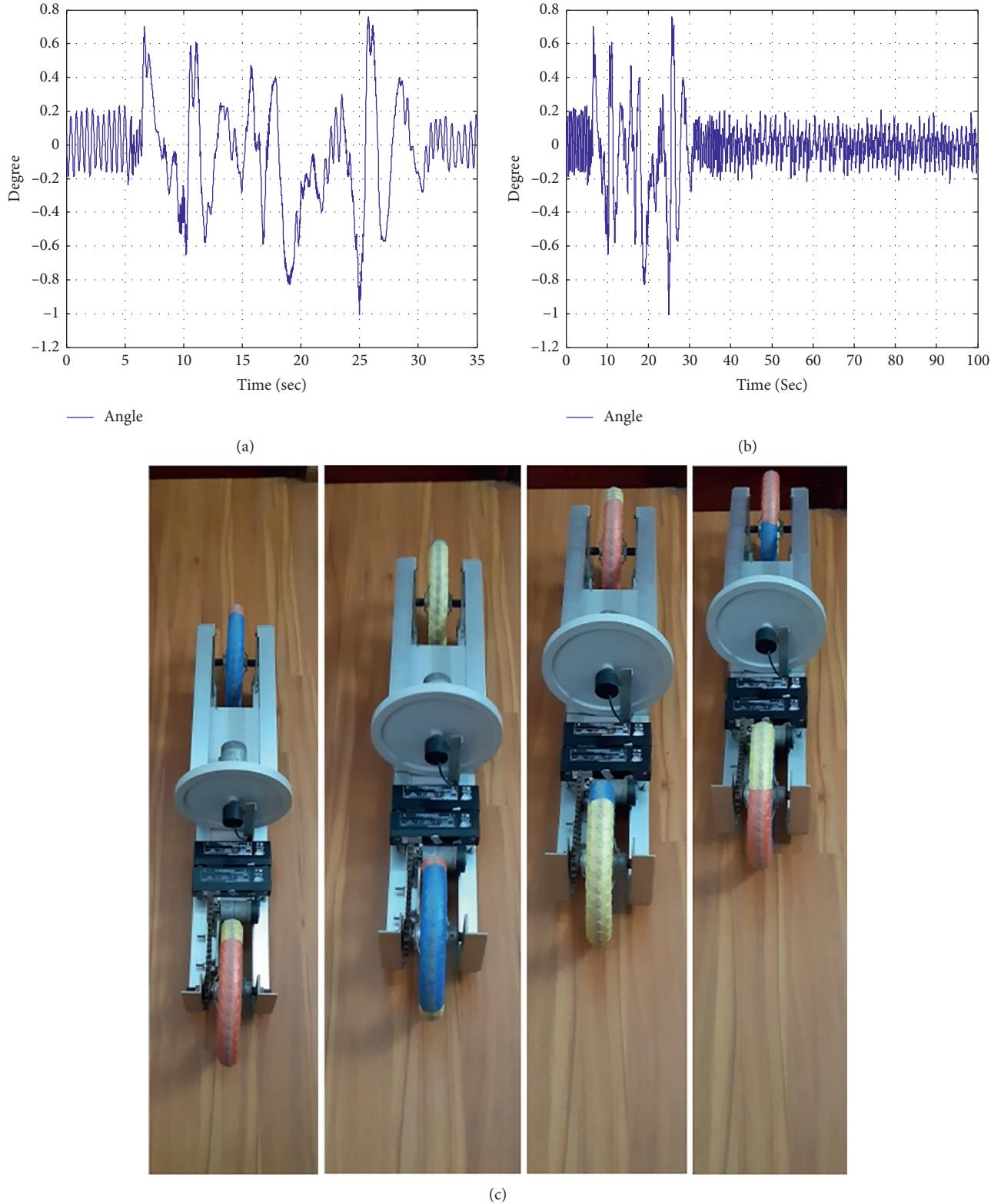


FIGURE 22: Experimental results when the vehicle is moving and carrying a load of 1.8 kg: (a) tilt angle response (0–35 seconds); (b) tilt angle response (0–100 seconds); (c) balanced vehicle.

moving. Thus, when the vehicle is moving, the angular oscillation amplitude increases, but balance and stability of the vehicle are still maintained.

5. Conclusions

The article introduces a self-balancing two-wheel vehicle model using flywheels according to the inverted pendulum principle. Modeling of the vehicle shows that the two-wheeled vehicle is a nonlinear and uncertain object. A high-order robust controller RH_{∞} is designed to maintain a stable balance of the two-wheel vehicle. To reduce the order of the controller, the article has used different order reduction algorithms. This study evaluated the two-wheel vehicle balance control system utilizing the reduced-order controllers and chose the most suitable low-order controller (2^{nd} -order controller) through the simulation. When used to control the balance of the two-wheel vehicle, a low-order control helps stabilize the vehicle when being stationary and moving and when the load changes. In the upcoming studies, we will apply optimal algorithms to find the low-order robust controller that meets the requirements of the self-balancing two-wheel control problem.

6. Future Recommendation Study

Research on solutions to increase the steady angle for self-balancing two-wheel vehicle; research on how to reduce the flywheel mass while maintaining the moment of inertia of the flywheel; research on how to improve the load-carrying capacity of self-balancing two-wheel vehicle; research on solutions to combine steering angle control with two-wheel vehicle balance control; comparing the efficiency of the robust control algorithm for the self-balancing two-wheel vehicle with other control algorithms are warranted.

Data Availability

This publication is supported by multiple datasets, which are available at locations cited in the reference section.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors gratefully acknowledge the Thai Nguyen University of Technology for supporting this work.

References

- [1] K. Astrom, R. Klein, and A. Lennartsson, "Bicycle dynamics and control," *IEEE Control Systems*, vol. 25, no. 4, pp. 26–47, 2005.
- [2] Y. Tanaka and T. Murakami, "Self sustaining bicycle robot with steering controller," in *Proceedings of the International Workshop on Advanced Motion Control*, pp. 193–197, Tokyo, Japan, March 2004.
- [3] T. Takei, M. Tsuruta, J. Okumura, T. Nakabayashi, and T. Tsubouchi, "Stabilized motion of a small sized bike robot only by steering control," in *Proceedings of the 11th IEEE International Workshop on Advanced Motion Control*, pp. 466–471, Nagaoka, Japan, March 2010.
- [4] S. Vatanashevapakorn and M. Parnichkun, "Steering control based balancing of a bicycle robot," in *Proceedings of the 2011 IEEE International Conference on Robotics and Biomimetics*, pp. 2169–2174, Karon Beach, Thailand, December 2011.
- [5] C. Huang, Y. Tung, and T. Yeh, "Balancing control of a robot bicycle with uncertain center of gravity," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5858–5863, Montreal, Canada, May 2017.
- [6] M. H. Shafiei and M. Emami, "Design of a robust path tracking controller for an unmanned bicycle with guaranteed stability of roll dynamics," *Systems Science & Control Engineering*, vol. 7, no. 1, pp. 12–19, 2019.
- [7] A. V. Beznos, A. M. Formalsky, E. V. Gurfinkel et al., "Control of autonomous motion of two-wheel bicycle with gyroscopic stabilization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2670–2675, Shanghai, China, July 1998.
- [8] J. M. Gallaspay, *Gyroscopic Stabilization of an Unmanned Bicycle*, M. S. Thesis, Auburn University, Auburn, AL, USA, 1999.
- [9] P. Y. Lam, T. K. Shin, and T. K. Sin, "Gyroscopic stabilization of a self-balancing robot bicycle," *International Journal of Automation Technology*, vol. 5, no. 6, pp. 916–923, 2011.
- [10] C. S. Stephen and R. G. Anouck, "Gyroscopic stabilisation of unstable vehicles: configurations, dynamics, and control," *Journal Vehicle System Dynamics, International Journal of Vehicle Mechanics and Mobility*, vol. 46, pp. 247–260, 2008.
- [11] B. T. Thanh and M. Parnichkun, "Balancing control of Bicyrobo by particle swarm optimization-based structure-specified mixed H2/Hoo control," *International Journal of Advanced Robotic Systems*, vol. 5, no. 4, pp. 395–402, 2008.
- [12] D. Karnopp, "Tilt control for gyro-stabilized two-wheeled vehicles," *Vehicle System Dynamics*, vol. 37, no. 2, pp. 145–156, 2002.
- [13] R. Lot and J. Fleming, "Gyroscopic stabilisers for powered two-wheeled vehicles," *Vehicle System Dynamics*, vol. 57, no. 9, pp. 1381–1406, 2019.
- [14] S. H. Park and S. Y. Yi, "Active balancing control for unmanned bicycle using scissored-pair control moment gyroscope," *International Journal of Control, Automation and Systems*, vol. 18, pp. 1–9, 2019.
- [15] S. Lee and W. Ham, "Self-stabilizing strategy in tracking control of unmanned electric bicycle with mass balance," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp. 2200–2205, Lausanne, Switzerland, December 2002.
- [16] M. Yamakita, A. Utano, and K. Sekiguchi, "Experimental study of automatic control of bicycle with balancer," in *Proceedings of the International Conference of Intelligent Robots and Systems*, pp. 5606–5611, Beijing, China, October 2006.
- [17] L. Keo, K. Yoshino, M. Kawaguchi, and M. Yamakita, "Experimental results for stabilizing of a bicycle with a flywheel balancer," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, October 2011.
- [18] Y. Kim, H. Kim, and J. Lee, "Stable control of the bicycle robot on a curved path by using a reaction wheel," *Journal of Mechanical Science and Technology*, vol. 29, no. 5, pp. 2219–2226, 2015.

- [19] S.-I. Lee, I.-W. Lee, M.-S. Kim, H. He, and J.-M. Lee, "Balancing and driving control of a bicycle robot," *Journal of Institute of Control, Robotics and Systems*, vol. 18, no. 6, pp. 532–539, 2012.
- [20] H. W. Kim, J. W. An, H. D. Yoo, and J. M. Lee, "Balancing control of bicycle robot using pid control," in *Proceedings of the International Conference on Control, Automation, and Systems*, Jeju, Korea, October 2013.
- [21] K. Kanjanawanishkul, "LQR and MPC controller design and comparison for a stationary self-balancing bicycle robot with a reaction wheel," *Kybernetika*, vol. 51, no. 1, pp. 173–191, 2015.
- [22] A. K. Vo, M. T. Nguyen, T. N. Nguyen et al., "PD controller for bicycle model balancing," *Robotica & Management*, vol. 23, no. 2, pp. 37–41, 2018.
- [23] C.-H. Chiu and C.-Y. Wu, "Bicycle robot balance control based on a robust intelligent controller," *IEEE Access*, vol. 8, pp. 84837–84849, 2020.
- [24] P. Seekhaoa, K. Tungpimolruth, and M. Parnichkuna, "Development and control of a bicycle robot based on steering and pendulum balancing," *Mechatronics*, vol. 69, 2020.
- [25] T.-D. Chu and C.-K. Chen, "Design and implementation of model predictive control for a gyroscopic inverted pendulum," *Applied Sciences*, vol. 7, no. 12, p. 1272, 2017.
- [26] S. Mobayen and F. Tchier, "Nonsingular fast terminal sliding-mode stabilizer for a class of uncertain nonlinear systems based on disturbance observer," *Scientia Iranica*, vol. 24, no. 3, pp. 1410–1418, 2017.
- [27] S. Mobayen and J. Ma, "Robust finite-time composite nonlinear feedback control for synchronization of uncertain chaotic systems with nonlinearity and time-delay," *Chaos, Solitons & Fractals*, vol. 114, pp. 46–54, 2018.
- [28] S. Afzal and P. Rajendra, "Reduced order modelling based control of two wheeled mobile robot," *Journal of Intelligent Manufacturing*, vol. 30, no. 3, pp. 1057–1067, 2019.
- [29] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice-Hall, Hoboken, NJ, USA, 1996.
- [30] A. Huseyin and G. Rahmi, "Robust H_∞ control of STMDs used in structural systems by hardware in the loop simulation method," *Actuators*, vol. 9, p. 55, 2020.
- [31] M. G. Ortega and F. R. Rubio, "Systematic design of weighting matrices for the H_∞ mixed sensitivity problem," *Journal of Process Control*, vol. 14, no. 1, pp. 89–98, 2004.
- [32] K. Benjelloun, E. K. Boukas, O. L. V. Costa, and P. Shi, "Design of robust controller for linear systems with Markovian jumping parameters," *Mathematical Problems in Engineering*, vol. 4, Article ID 219179, 20 pages, 1997.
- [33] Z. Lu, L. Wang, F. Zhang, and F. Xu, "The design of robust controller for networked control system with time delay," *Mathematical Problems in Engineering*, vol. 2015, Article ID 189013, 8 pages, 2015.
- [34] C. H. Nguyen, K. N. Vu, and D. H. Dao, "Applying order reduction model algorithm for balancing control problems of two-wheeled mobile robot," in *Proceedings of the 8th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1302–1307, Melbourne, Australia, June 2013.
- [35] D. Youla, J. Bongiorno Jr., and H. Jabr, "Modern Wiener–Hopf design of optimal controllers Part I: the single-input-output case," *IEEE Transactions on Automatic Control*, vol. 21, no. 1, pp. 3–13, 1976.
- [36] J. Ma, S. Wang, X. Gao, X. Zhu, Y. Qiu, and Z. Wang, "Youla parameterization robust control strategy considering power system uncertainties," *Electric Power Components and Systems*, vol. 42, no. 11, pp. 1152–1157, 2014.
- [37] D.-W. Gu, P. H. Petkov, and M. M. Konstantinov, *Robust Control Design with MATLAB®*, Springer, London, UK, 2nd edition, 2013.
- [38] C. H. Nguyen, K. N. Vu, and H. T. Do, "Model reduction based on triangle realization with pole retention," *Applied Mathematical Sciences*, vol. 9, no. 44, pp. 2187–2196, 2015.
- [39] B. Moore, "Principal component analysis in linear systems: controllability, observability, and model reduction," *IEEE Transactions on Automatic Control*, vol. 26, no. 1, pp. 17–32, 1981.
- [40] M. G. Safonov and R. Y. Chiang, "A schur method for balanced-truncation model reduction," *IEEE Transactions on Automatic Control*, vol. 34, no. 7, pp. 729–733, 1989.
- [41] K. Glover, "All optimal Hankel-norm approximations of linear multivariable systems and their L_∞-error bounds†," *International Journal of Control*, vol. 39, no. 6, pp. 1115–1193, 1984.
- [42] A. Antoulas, *Approximation of Large-Scale Dynamical System*, SIAM, Philadelphia, PA, USA, 2005.
- [43] K. P. Arvind and P. Rajendra, "Order reduction in linear dynamical systems by using improved balanced realization technique," *Circuits, Systems, and Signal Processing*, vol. 38, pp. 5289–5303, 2019.
- [44] M. Green, "Balanced stochastic realizations," *Linear Algebra and Its Applications*, vol. 98, pp. 211–247, 1989.
- [45] E. A. Jonckheere, P. Harshavaradhana, and L. M. Silverman, "Stochastic balancing and approximation-stability and minimality," *IEEE Transactions on Automatic Control*, vol. 29, pp. 744–746, 1984.
- [46] K. Zhou, "Frequency-weighted model reduction with ∞ error bounds," *Systems & Control Letters*, vol. 21, no. 2, pp. 115–125, 1993.
- [47] M. G. Safonov and R. Y. Chiang, "Model reduction for robust control: a schur relative error method," *International Journal of Adaptive Control and Signal Processing*, vol. 2, no. 4, pp. 259–272, 1988.
- [48] K. Zhou, G. Salomon, and E. Wu, "Balanced realization and model reduction for unstable systems," *International Journal of Robust and Nonlinear Control*, vol. 9, no. 3, pp. 183–198, 1999.
- [49] M. Vidyasagar, *Control System Synthesis-A Factorization Approach*, The MIT Press, London, UK, 1985.
- [50] A. Zilochian, "Balanced structures and model reduction of unstable systems," in *IEEE Proceedings of Southeastcon '91*, pp. 1198–1201, Williamsburg, VA, USA, April 1991.