Robust Optimal Controller for Two-wheel Self-Balancing Vehicles Using Particle Swarm Optimization

Vu Ngoc Kien[1], Nguyen Tien Duy[2], Dao Huy Du[2], Nguyen Phuong Huy[2], Nguyen Hong Quang[3]*

1 Faculty of Electrical Engineering, Thai Nguyen University of Technology, Thai Nguyen City, Vietnam
[2] Faculty of Electronics Engineering, Thai Nguyen University of Technology, Thai Nguyen City, Vietnam
[3] Faculty of Mechanical, Electrical, Electronics Technology, Thai Nguyen University of Technology, Thai Nguyen City, Vietnam
*Correspondence: quang.nguyenhong@tnut.edu.vn

***Abstract*—Control of a self-balancing vehicle is a challenging but exciting research topic. The challenge of researching self-balancing bicycles is maintaining balance when the bike is stationary and when the bike is moving. This paper, through analysis and comparison of two-wheeled vehicle balancing methods, shows that the method that best meets the requirements of the two-wheeled vehicle balance control problem is the balancing method using a flywheel stabilizer. Compared with the gyroscopic flywheel stabilizer, the inverted pendulum flywheel stabilizer has the advantages of fast response speed and energy saving, so we choose the pendulum flywheel stabilizer to reverse to control the balance of the two-wheeler. By modeling and analyzing the two-wheel vehicle model, it shows that the vehicle model is subjected to uncertainties. Hence, the robust controller is an appropriate controller for balancing two-wheel vehicles. However, the controller designed according to the robust control algorithm RH_∞ is often high-order, affecting the actual control quality. We proposed using the particle swarm optimization (PSO) algorithm to find a low-order robust controller from the high-order robust controller. By comparing the efficiency of the low-order robust controller according to PSO with the high-order robust controller and other low-order robust controllers, we have proven the correctness of the low-order robust controller according to PSO. Simulation results show that a two-wheel vehicle using a low-order robust controller according to PSO can stabilize the vehicle and give good control quality.**

***Keywords*—Two-wheel self-balancing vehicles, Inverted pendulum, Robust controller RH∞, article swarm optimization, Model order reduction**

I. Introduction

Bicycles (two-wheel vehicles) are a widely used means of transport worldwide. It is pretty challenging to balance and stabilize the bicycle without a driver. Therefore, the self-balancing bicycle is an exciting area of research for researchers. The equilibrium state of the bicycle depends on its speed [1]. When the bicycle is stationary, it is considered an unstable model. The bicycle can achieve stable equilibrium [1] under certain conditions when it is moving forward [1]. Different models are used to model bicycles, such as linear and nonlinear models [1]. Many other solutions have been proposed to solve the balancing problem of bicycle without a driver, in which it can be categorized into two groups: with or without a stabilizer.

Controlling the vehicle's steering angle [2-6] is the most typical method of balancing a two-wheel vehicle without a stabilizer. The control system will change the vehicle's steering angle in conjunction with the vehicle's moving state to generate centrifugal force to maintain the vehicle's equilibrium. This method has the advantage that there is no need to add a stabilizer, but the downside is that it is impossible to maintain balance while the vehicle is stationary.

There are many solutions to the design of stabilizers for self-balancing two-wheelers. This solution can use a flywheel in studies [7-11] or use two flywheels in studies [12-14]. The stabilizer uses a flywheel according to the gyroscope principle [7-14]. In this method, the flywheel rotates at high speed to accumulate energy; then, the control system will control the angle of the flywheel to generate force that helps maintain the vehicle's equilibrium. The advantages of this method are fast response and a large balancing force.

A balancer in the form of a repositionable weight has been introduced in [15]. In [15], by adding weight and changing the position of the weight on the vehicle, the vehicle's center of gravity is changed to create a state of equilibrium for the vehicle. This method helps the vehicle maintain equilibrium when stationary and when the vehicle is in motion. But adding heavy objects to the vehicle, it will increase the size and weight of the vehicle and the response speed of the system is often slow. To help the system have a faster response speed, in [16-17], replace the weight with a flywheel. By changing the flywheel's structure, the vehicle's center of gravity also changes, thereby helping the vehicle maintain a state of balance.

Manuscript received August 9, 2022; revised September 1, 2022; accepted October 5, 2022.

doi: 10.18178/ijmerr.12.1.16-22

In [18-23], the balancer uses a flywheel based on the inverted pendulum principle. By controlling the rotation of the flywheel, the control system will create a force to balance the gravity of the vehicle, which will help the vehicle balance stably. The basic feature of this method is that the flywheel speed is slow, so the power consumption of the flywheel is low, and the response speed of the control system is fast. However, the value of the balancing force of the system is not large, so the range of tilt angle to maintain the balance of the vehicle is small. To expand the range of tilt angle, the study [24] combined the center of gravity change method and the method of steering angle control.

This paper uses the flywheel based on the inverted pendulum principle to build a self-balancing two-wheel model. In terms of control, there are many different methods, such as PD control [21], PID control [17-19], LQG and MPC control [21], and robust control [22]. The analysis of balancing solutions for two-wheel vehicles shows that each method has its advantages and disadvantages. When the vehicle is operating in reality, the vehicle will be affected by some uncertain factors, such as load, noise, external force, etc., so the vehicle model can be considered an uncertain model [10]. Therefore, robust optimal control is a potential solution.

This study introduces a two-wheel vehicle balance control system using a robust optimal algorithm. However, the stable optimal controller usually has high order [27-32].

Design a low-order stable controller for two-wheeled vehicles proposed in the reference [11]. It was suggested to use the PSO algorithm to find a low-order controller satisfying the requirements of sustainable control. In the study [30-33], a high-order controller was designed to meet the needs of sustainable rules. Then it was proposed to use the pole-conserving order reduction algorithm to simplify the controller. The study [34] suggests using algorithms based on balanced truncation and all-order reduction for high-order controllers.

Considering the problem of model order reduction, in addition to order reduction algorithms, it is also possible to use optimal order reduction algorithms. However, in the problem of order reduction of the high-order optimal controller for self-balancing two-wheelers, there have been no studies using optimal order reduction algorithms. Many optimization algorithms can be used to determine the optimal low-order controller, such as Genetic Algorithm (GA), PSO, and Ant colony optimization (ACO) algorithm... The advantage of the PSO algorithm is the ability to search within a given availability area. The PSO algorithm does not require a detailed mathematical description, and it is easy to find the best possible value by optimizing the objective function and high computational efficiency. The disadvantage of the PSO algorithm is that the accuracy is not high, and the sensitivity is different.

Therefore, in this paper, we propose to use the PSO algorithm to reduce the order of the high-order stable controller of the two-wheeled vehicle balance control system. Identifying the low-level controller (by the PSO algorithm) will simplify the control programming code, reduce the response time, and help meet the system's real-time control requirements.

The layout of the article is as follows: Mathematical models of vehicles and high-order robust controllers are presented in Section 2. Next, Section 3 introduces the PSO algorithm and the steps to determine the parameters of the low-order robust controller according to the PSO algorithm. Then, section 4 presents the simulation results of the control system using reduced-order controllers and the original controller. Section 5 is the conclusion of the paper.

II. Dynamic Model, Mathematical Model, and Robust Control of Two-Wheel Self-Balancing Vehicles

A. Dynamic Model and Mathematical Model of Two-wheel Self-balancing Vehicles

A bicycle model designed to be able to move forward and backward, maintaining self-equilibrium in [33] is shown in Fig. 1 as follows:

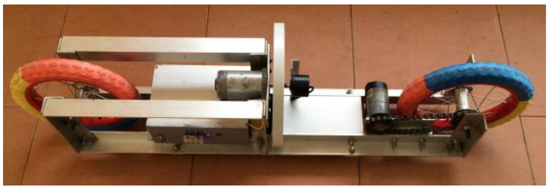

Figure 1. Self-balancing two-wheel vehicle model [36]

The vehicle's balance system is built on the principle of an inverted pendulum. Accordingly, the electric motor will provide motion to the flywheel so that the flywheel rotates around the axis with acceleration α to create torque. This torque will balance with gravity to help the vehicle maintain equilibrium. Details of the vehicle configuration are presented in [33].

Using the Lagrange equation [8] and the linearization method around the equilibrium point ($\alpha = \delta = 0$, $\sin\alpha = \alpha$), with the assumption that the tilt angle of the vehicle is minimal ($\alpha < 10^0$), the model of the vehicle is described as follows [33]:

$$I_f\ddot{\alpha} + I_f\ddot{\delta} = T_m\left[\frac{U - T_e\dot{\delta}}{R}\right]$$

Define $x = \begin{bmatrix} \alpha = x_1 \\ \dot{\alpha} = x_2 \\ \dot{\delta} = x_3 \end{bmatrix}; y = \dot{x}; u = U$

and $E = \left(m_b h_b^2 + m_f h_f^2 + I_b + I_f\right)$; $F = \left(m_b h_b + m_f h_f\right)$

The state model of the vehicle has the following form[33]:

$$\dot{x} = \mathbf{A}x + \mathbf{B}u$$
$$y = \mathbf{C}x + \mathbf{D}u$$

where

$$\mathbf{A}=\begin{bmatrix} 0 & 1 & 0 \\ \dfrac{Fg}{(E-I_f)} & 0 & \dfrac{aT_mT_e}{R(E-I_f)} \\ -\dfrac{Fg}{(E-I_f)} & 0 & -aT_mT_e\dfrac{E}{I_fR(E-I_f)} \end{bmatrix}$$

$$\mathbf{B}=\begin{bmatrix} 0 \\ -\dfrac{aT_m}{R\left(E-I_f\right)} \\ aR_m\dfrac{E}{I_fR(E-I_f)} \end{bmatrix}.$$

$$\mathbf{C}=\begin{bmatrix}1 & 0 & 0\end{bmatrix};D=\begin{bmatrix}0\end{bmatrix}.$$

Applying the vehicle's parameters in Table I and transforming the state space model into a transfer function representation, we get:

TABLE I. THE PARAMETERS OF VEHICLE MODEL [33]

Parameters	Value	Unit	Meaning
I_b	0.1105	Kg.m2	Moment of inertia of the vehicle
h_b	0.105	m	Height of center of gravity of the vehicle
I_f	0.03289	Kg.m2	Moment of inertia of the flywheel
h_f	0.205	m	Height of center of gravity of the flywheel
m_b	10.024	Kg	Mass of the vehicle (excluding flywheel)
m_f	3.976	Kg	Mass of the flywheel
T_e	0.045	V.s/Rad	Electric motor constant
T_m	0.045	Nm/A	Motor torque constant
R	0.52	Ω	Motor resistance
g	9.81	m/s2	Gravity acceleration

$$\mathbf{W}(s)=\frac{\boldsymbol{\alpha}(s)}{\mathbf{U}(s)}=\frac{-0.223s}{s^3+0.1284s^2-47.2s-5.589} \quad (1)$$

B. *Robust Controller for Two-wheel Self-balancing Vehicles*

The structure of the vehicle balance control system is shown in Fig. 2 as follows:

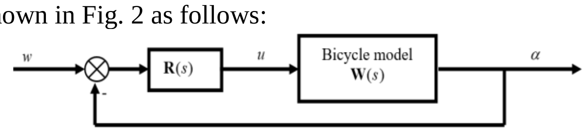

Figure 2. The structure of the vehicle balance control system [33]

Designing a powerful controller according to the control design process in [31-32] for the vehicle balancing system, we obtain the following controller:

$$\mathbf{R}(s)=\frac{\mathbf{A}(s)}{\mathbf{B}(s)} \quad (*)$$

$$\begin{aligned}
\mathbf{A}(s)&=-2.23.10^{-7}s^{30}-4.67.10^{-4}s^{29}-0.266s^{28}\\
&-22.96s^{27}-1006s^{26}-2.853.10^{4}s^{25}\quad-5.837.10^{5}s^{24}\\
&-4.199.10^{11}s^{18}-9.144.10^{6}s^{23}-1.139.10^{8}s^{22}\\
&-1.158.10^{9}s^{21}-9.776.10^{9}s^{20}-6.949.10^{10}s^{19}\\
&-2.172.10^{12}s^{17}-9.663.10^{12}s^{16}-3.71.10^{13}s^{15}\\
&-1.231.10^{14}s^{14}-3.53.10^{14}s^{13}-8.74.10^{14}s^{12}\\
&-1.862.10^{15}s^{11}-3.398.10^{15}s^{10}\\
&-5.276.10^{15}s^{9}-6.903.10^{15}s^{8}-7.511.10^{15}s^{7}\\
&-6.676.10^{15}s^{6}-4.721.10^{15}s^{5}\\
&-2.556.10^{15}s^{4}-9.953.10^{14}s^{3}-2.482.10^{14}s^{2}\\
&-2.977.10^{13}s-0.00439
\end{aligned}$$

$$\begin{aligned}
\mathbf{B}(s)&=4.971.10^{-14}s^{30}+2.032.10^{-10}s^{29}\\
&+2.663.10^{-7}s^{28}+1.221.10^{-4}s^{27}+9.72.10^{-3}s^{26}\\
&+0.3918s^{25}+10.14s^{24}+187.1s^{23}+2612s^{22}\\
&+2.862.10^{4}s^{21}+1.088.10^{7}s^{18}\\
&+2.523.10^{5}s^{20}+1.82.10^{6}s^{19}+5.428.10^{7}s^{17}\\
&+2.273.10^{8}s^{16}+8.005.10^{8}s^{15}\\
&+2.372.10^{9}s^{14}+5.9.10^{9}s^{13}+1.225.10^{10}s^{12}\\
&+2.107.10^{10}s^{11}+2.962.10^{10}s^{10}\\
&+3.341.10^{10}s^{9}+2.941.10^{10}s^{8}+1.931.10^{10}s^{7}\\
&+8.743.10^{9}s^{6}+2.286.10^{9}s^{5}\\
&+1.519.10^{8}s^{4}-5.226.10^{7}s^{3}\\
&+3.6.10^{-6}s^{2}+5.32.10^{-22}s
\end{aligned}$$

Because the programming code of the 30th-order controller is complex, the control system will have a slow response rate which may cause the vehicle to fail to maintain equilibrium. To simplify the program code, we try simplifying the controller of order 30. The objective of the controller simplification problem is: the controller's order is as small as possible; The control system when using a low-level controller, still ensures stable vehicle balance.

For the controller simplification problem, there are two basic solutions as follows:

Solution 1: Determine the low-order controller from the high-order controller using the order reduction algorithm. The selected low-order controller will be the controller with the lowest order in which the control system, when using the controller, still meets the requirements of maintaining vehicle stability.

Solution 2: Apply optimization algorithms to determine the parameters of a low-order controller with a fixed structure so that the low-order controller's response coincides with the high-order controller's response.

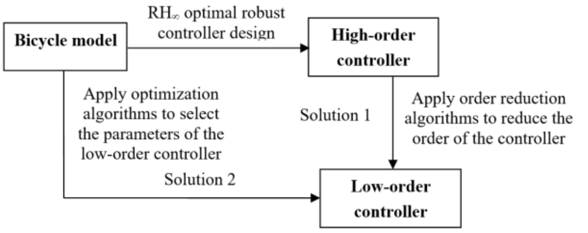

Figure 3. Two solutions for designing low-order controllers for bicycle model

In this paper, we choose Solution 2 to simplify the root controller. We compare and evaluate the low-order controllers by both methods to choose a suitable low-order controller.

III. Order Reduction of the High-order Robust Controller Using the PSO

Swarm optimization is a random search algorithm based on simulating the behavior and interaction of flocks of birds or schools of fish looking for food [38]. Each bird (or individual element) in the flock (population) is characterized by two components, the position vector x_i and the velocity vector (displacement) v_i. Each individual has a fitness value evaluated by the fitness function. Initially, the PSO is initialized with random position and velocity vectors. Then, in each iteration of the algorithm, the velocity the position vectors of each individual will be updated according to formulas (5) and (6). Also, at each iteration (every time moving in d-dimensional space), each individual is affected by two pieces of information.

The first information is the best position it has achieved so far, P_{best_id} . The second information is the best position in all the search processes of individuals in the population so far, G_{best}. The modeling of updating each individual's position according to its best position and that of all individuals in the population up to the present time is illustrated in Fig. 4.

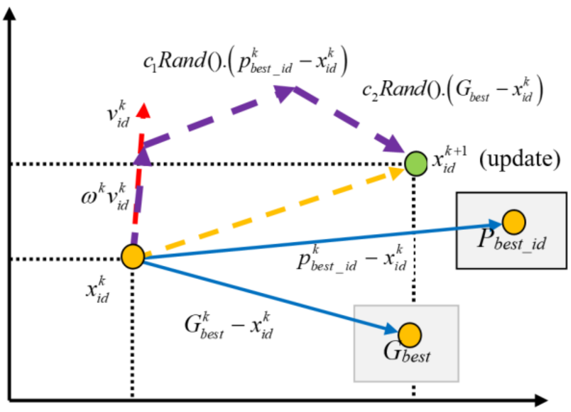

Figure 4. Graph illustrating a search point using PSO

$$V_{id}^{k+1} = \omega^k * V_{id}^k + C_1 * Rand() * \left(P_{best_i d} - x_{id}^k\right) + C_2 * Rand() * \left(G_{best} - x_{id}^k\right) \quad (2)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (3)$$

$$\omega^k = \omega_{max} - \frac{k*(\omega_{max} - \omega_{min})}{iter\_max} \quad (4)$$

Each $id-\mathrm{th}$ individual is a vector $P_i = \left[p_{id}^1, p_{id}^2, \ldots, p_{id}^d\right]$ and its corresponding P_{best_id} is calculated as below:

$$P_{(best\_id)}^{(k+1)} = f(x) = \begin{cases} (P^k_{(best_i d)}, \text{if fitness}(x_i d^{(k+1)}) > P^k_{(best_{id})} \\ \text{fitness}(x_i d^{(k+1)}), \text{ if fitness}(x_i d^{(k+1)}) \le P^k_{(best_{id})}) \end{cases} \quad (5)$$

The value of G_{best} at the iteration k is:

$$G_{best} = \min\left(P^k_{best_{id}}\right) \quad (6)$$

In which:

X^k_{id} and V^k_{id} is the current position and velocity of individual id at iteration k. The limit of velocity is defined in the domain [-Vmax, Vmax], where V_{max} is a user-defined constant;

ω is the time-varying inertia weight, chosen as presented in [38], which linearly decreases from 0.9 to 0.4 according to (4).

c1 and c2 are two learning factors used to control the influence of social and cognitive components, defined as [38], i.e., c1=c2=2.

PSO-based order reduction

In this study, we reduce the order of the transfer function (*) from 30 to 4, that is, A(s) and B(s) have a maximum degree of 4.

The coefficients of A(s) and B(s) are a_i and b_i, respectively. The fourth order transfer function has the following form:

$$H(s) = \frac{a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0}{b_4 s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0} \quad (**)$$

According to (**), ten variables need to be optimized. The trial and error method determines the search domain for these variables. The objective function is selected according integrates the absolute error (IAE) index as

$$\text{fitness} = \sum_{k=1}^{n} |e(k)| \rightarrow \min \quad (7)$$

where e(k) is the error at time sample k,n is the total number of data samples of a simulation run.

The particle swarm() function has been supported for optimization in Matlab Toolbox according to the PSO algorithm. In this study, we use the existing particle swarm() process and follow the following algorithm:

We obtained the following results by conducting a parameter search of the 4th-order controller according to the flowchart of Fig. 5:

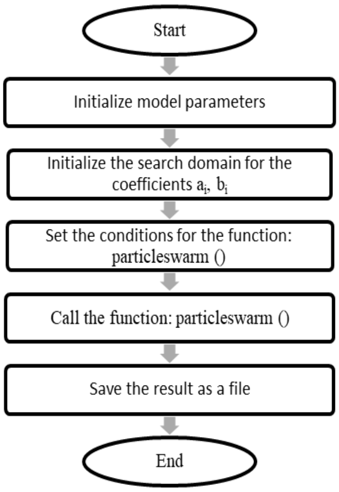

Figure 5. Flowchart of implementation according to PSO algorithm

a_4= -1.498 ×10^6; a_3= -3.54567 ×10^7; a_2= -7.32 ×10^8 ; a_1= -3.66749 ×10^8; a_0= -1.51109 ×10^7

b_4= 1.1966; b_3= 471.218; b_2= 12252; b_1= -872.448; b_0= 987.251

Substituting the coefficients a_i and b_i into the formula (**), we obtain the transfer function of the 4th-order controller as follows:

$$H(s) = \frac{-1.498\times10^{6}s^{4} - 3.54567\times10^{7}s^{3} - 7.32\times10^{8}s^{2} - 3.66749\times10^{8}s - 1.51109\times10^{7}}{1.1966s^{4} + 471.218s^{3} + 12252s^{2} - 872.448s + 987.251}$$

IV. PSO Controlling a Two-wheel Self-balancing Vehicle Using a Robust Low-level Controller

The response of the two-wheel self-balancing vehicle control system is shown in Fig. 6 and Fig. 7.

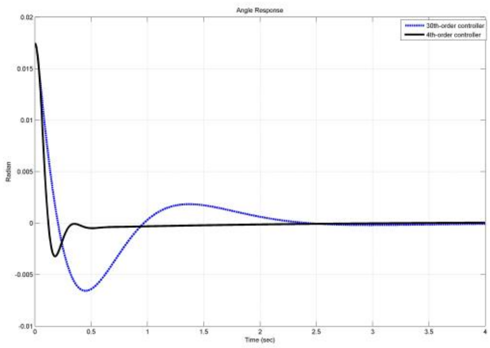

Figure 6. Tilt angle response of the vehicle control system using the 4th order controller according to the PSO algorithm and the 30th order controller

Comment: From Fig. 6, the 4th-order controller, according to the PSO algorithm, can balance the two-wheel vehicles. The vehicle control system using the 4th-order controller has shorter transient times and less oscillation than the control system using the original controller. However, the overshoot of the control system using the 4th-order controller is larger than that of the vehicle control system using the original controller.

To evaluate the efficiency of the reduce-order controller, we simplify the high-order controller according to Solution 1 by the balanced truncation algorithm (balancmr) [35,37] and the balanced stochastic truncation algorithm (bstmr) [36].

TABLE II. 4TH-Order Controllers According to Solution 1

Order reduction algorithm	The transfer function of the reduced order controller – $R_r(s)$
Balanced truncation model reduction (balancmr) in [35,37]	$\frac{-4.531\times10^{4}s^{4} - 8.686\times10^{4}s^{3} - 4.417\times10^{5}s^{2} - 4.32\times10^{5}s - 1.986\times10^{5}}{s^{4} + 199.9s^{3} - 158.7s^{2} - 2.782\times10^{-14}s - 2.132\times10^{-17}}$
Balanced stochastic truncation model reduction (bstmr) in [36]	$\frac{-4.531.10^{4}s^{4} - 7.535.10^{4}s^{3} - 4.479.10^{5}s^{2} - 4.342.10^{5}s - 1.985.10^{5}}{s^{4} + 199.5s^{3} - 158.4s^{2} - 3.556.10^{-14}s - 1.31.10^{-15}}$

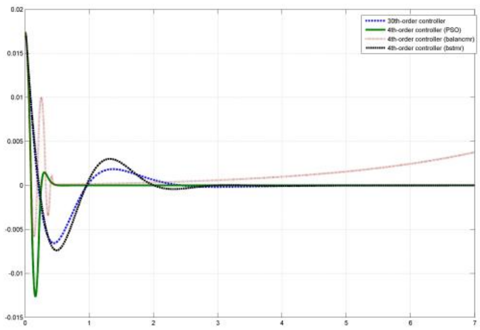

Figure 7. Tilt response of the vehicle control system using 4th and 30th-order controls

Comment: From Fig. 7, it can be seen that the control system using the 4th order controller according to the balanced truncation algorithm cannot maintain the vehicle balance. The control system uses a 4th-order controller according to the balanced stochastic truncation algorithm, and the PSO algorithm is capable of maintaining vehicle balance.

The control system using the 4th order controller (PSO algorithm) has better response quality than the control system using the 4th order controller (bstmr algorithm).

By comparing and evaluating the reduced order controllers according to Solution 1 and Solution 2, we choose the 4th order controller according to the PSO algorithm to replace the original controller.

V. Conclusion

Designing a robust controller for a self-balancing two-wheel vehicle balancing system using flywheels according to the principle of an inverted pendulum obtained a 30th-order controller. Using a 4th-order controller to control the two-wheeled vehicle balancing system shows that the system ensures stable balance. Using the PSO optimization algorithm, we determined that a 4th-order controller can replace a 30th-order controller. The simulation results also show that: The response quality of the control system when using the 4th-order controller, according to PSO, is better than when using the 30th-order controller and other 4th-order controllers. The paper results show that the solution using the PSO optimization algorithm to determine the parameters of the low-order controller is feasible and highly effective compared to other order reduction algorithms. The contribution of this paper is to develop a method to design a low-order optimal controller for a two-wheeled vehicle balance control system using the PSO algorithm. In the next study, we will use many different optimization algorithms to identify low-order controllers and evaluate the efficiency of low-order controllers in accurate control systems.

CONFLICT OF INTEREST

The authors declare no conflict of interest

AUTHOR CONTRIBUTIONS

V. N. K. was originally responsible for conceptualization and methodology. N.T.D., D.H.D., and N.P.H. verified the simulation data, data curation, investigation, and draft writing. N.H.Q. contributes a formal analysis of review, editing, and supervision. All authors have read and agreed to the published version of the manuscript.

ACKNOWLEDGMENT

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

REFERENCES

[1] K. J. Astrom, R. E. Klein, and A. Lennartsson, "Bicycle dynamics and control: Adapted bicycles for education and research," *IEEE Control Syst.*, vol. 25, no. 4, pp. 26–47, 2005, doi: 10.1109/MCS.2005.1499389.

[2] Y. Tanaka and T. Murakami, "Self sustaining bicycle robot with steering controller," in *Proc. International Workshop on Advanced Motion Control, AMC*, 2004, pp. 193–197, doi: 10.1109/amc.2004.1297665.

[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 *Proc. International Workshop on Advanced Motion Control, AMC*, 2010, pp. 466–471, doi: 10.1109/AMC.2010.5464085.

[4] S. Vatanashevanopakorn and M. Parnichkun, "Steering control based balancing of a bicycle robot," in *Proc. 2011 IEEE International Conference on Robotics and Biomimetics, ROBIO*, 2011, pp. 2169–2174, doi: 10.1109/ROBIO.2011.6181613.

[5] C. F. Huang, Y. C. Tung, and T. J. Yeh, "Balancing control of a robot bicycle with uncertain center of gravity," in *Proc. IEEE International Conference on Robotics and Automation*, 2017, pp. 5858–5863, doi: 10.1109/ICRA.2017.7989689.

[6] X. Li, H. Dong, F. Han, and J. Li, "A survey on set-membership filtering for networked control systems under communication protocols," *Syst. Sci. Control Eng.*, vol. 6, no. 1, pp. 293–303, 2018, doi: 10.1080/21642583.2018.1494639.

[7] A. V. Beznos *et al.*, "Control of autonomous motion of two-wheel bicycle with gyroscopic stabilisation," in *Proc. IEEE International Conference on Robotics and Automation*, 1998, vol. 3, pp. 2670–2675, doi: 10.1109/ROBOT.1998.680749.

[8] H. Yetkin, S. Kalouche, M. Vernier, G. Colvin, K. Redmill, and U. Ozguner, "Gyroscopic stabilization of an unmanned bicycle," in *Proc. the American Control Conference*, 2014, pp. 4549–4554, doi: 10.1109/ACC.2014.6859392.

[9] P. Y. Lam, "Gyroscopic stabilization of a kid-size bicycle," in *Proc. 2011 IEEE 5th Int. Conf. Cybern. Intell. Syst. CIS 2011*, vol. 5, no. 6, pp. 247–252, 2011, doi: 10.1109/ICCIS.2011.6070336.

[10] S. C. Spry and A. R. Girard, "Gyroscopic stabilisation of unstable vehicles: Configurations, dynamics, and control," *Veh. Syst. Dyn.*, vol. 46, no. SUPPL.1, pp. 247–260, 2008, doi: 10.1080/00423110801935863.

[11] B. T. Thanh and M. Parnichkun, "Balancing control of bicyrobo by particle swarm optimization-based structure-specified mixed H2/H∞ control," *Int. J. Adv. Robot. Syst.*, vol. 5, no. 4, pp. 395–402, 2008, doi: 10.5772/6235.

[12] D. Karnopp, "Tilt control for gyro-stabilized two-wheeled vehicles," *Veh. Syst. Dyn.*, vol. 37, no. 2, pp. 145–156, 2002, doi: 10.1076/vesd.37.2.145.3535

[13] R. Lot and J. Fleming, "Gyroscopic stabilisers for powered two-wheeled vehicles," *Veh. Syst. Dyn.*, vol. 57, no. 9, pp. 1381–1406, 2019, doi: 10.1080/00423114.2018.1506588

[14] S. H. Park and S. Y. Yi, "Active balancing control for unmanned bicycle using scissored-pair control moment gyroscope," *Int. J. Control. Autom. Syst.*, vol. 18, no. 1, pp. 217–224, 2020, doi: 10.1007/s12555-018-0749-7.

[15] S. Lee and W. Ham, "Self stabilizing strategy in tracking control of unmanned electric bicycle with mass balance," in *Proc. IEEE International Conference on Intelligent Robots and Systems*, 2002, vol. 3, pp. 2200–2205, doi: 10.1109/irds.2002.1041594.

[16] M. Yamakita, A. Utano, and K. Sekiguchi, "Experimental study of automatic control of bicycle with balancer," in *Proc. IEEE International Conference on Intelligent Robots and Systems*, 2006, pp. 5606–5611, doi: 10.1109/IROS.2006.282281.

[17] L. Keo, K. Yoshino, M. Kawaguchi, and M. Yamakita, "Experimental results for stabilizing of a bicycle with a flywheel balancer," in *Proc. IEEE International Conference on Robotics and Automation*, 2011, pp. 6150–6155, doi: 10.1109/ICRA.2011.5979991.

[18] Y. Kim, H. Kim, and J. Lee, "Stable control of the bicycle robot on a curved path by using a reaction wheel," *J. Mech. Sci. Technol.*, vol. 29, no. 5, pp. 2219–2226, 2015, doi: 10.1007/s12206-015-0442-1.

[19] S. I. Lee, I. W. Lee, M. S. Kim, H. He, and J. M. Lee, "Balancing and driving control of a bicycle robot," *J. Inst. Control. Robot. Syst.*, 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 *International Conference on Control, Automation and Systems*, 2013, pp. 145–147, doi: 10.1109/ICCAS.2013.6703879.

[21] Y. Kim, H. Kim, and J. Lee, "Stable control of the bicycle robot on a curved path by using a reaction wheel," *J. Mech. Sci. Technol.*, vol. 29, no. 5, pp. 2219–2226, 2015, doi: 10.1007/s12206-015-0442-1.

[22] A. K. Vo *et al.*, "Pd controller for bicycle model balancing," *Robotica*, vol. 2, no. 1, pp. 37–42, 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, doi: 10.1109/ACCESS.2020.2992792.

[24] P. Seekhao, K. Tungpimolrut, and M. Parnichkun, "Development and control of a bicycle robot based on steering and pendulum balancing," *Mechatronics*, vol. 69, p. 102386, 2020, doi: 10.1016/j.mechatronics.2020.102386.

[25] A. Sikander and R. Prasad, "Reduced order modelling based control of two wheeled mobile robot," *J. Intell. Manuf.*, vol. 30, no. 3, pp. 1057–1067, 2019, doi: 10.1007/s10845-017-1309-3.

[26] N. K. Vu, H. Q. Nguyen, K. T. Ngo, and P. N. Dao, "Study on model reduction algorithm based on schur analysis," 2021, pp. 179–188, doi: 10.1007/978-981-15-7527-3_18.

[27] H. Aggumus and R. Guclu, "Robust H∞ control of STMDs used in structural systems by hardware in the loop simulation method," in *Actuators*, 2020, vol. 9, no. 3, p. 55, doi: 10.3390/ACT9030055.

[28] M. G. Ortega and F. R. Rubio, "Systematic design of weighting matrices for the H∞ mixed sensitivity problem," *J. Process Control*, vol. 14, no. 1, pp. 89–98, 2004, doi: 10.1016/S0959-1524(03)00035-0.

[29] Z. Lu, L. Wang, F. Zhang, and F. Xu, "The design of robust controller for networked control system with time delay," *Math. Probl. Eng.*, vol. 2015, 2015, doi: 10.1155/2015/189013.

[30] 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 *Proc. the 2013 IEEE 8th Conference on Industrial Electronics and Applications, ICIEA 2013*, 2013, pp. 1302–1307, doi: 10.1109/ICIEA.2013.6566568.

[31] J. Ma, S. Wang, X. Gao, X. Zhu, Y. Qiu, and Z. Wang, "Youla parameterization robust control strategy considering power system uncertainties," *Electr. Power Components Syst.*, vol. 42, no. 11, pp. 1152–1157, 2014, doi: 10.1080/15325008.2014.921948.

[32] D. W. Gu, P. H. Petkov, and M. M. Konstantinov, *Robust Control Design with MATLAB®*, no. 9781447146810. Springer Science & Business Media, 2013.

[33] N. K. Vu and H. Q. Nguyen, "Balancing control of two-wheel bicycle problems," *Math. Probl. Eng.*, vol. 2020, 2020, doi: 10.1155/2020/6724382.

[34] Ngoc Kien Vu, Hong Quang Nguyen, "Design low-order robust controller for self-balancing two-wheel vehicle," *Mathematical Problems in Engineering*, vol. 2021, p. 22, 2021. https://doi.org/10.1155/2021/6693807

[35] N. K. Vu and H. Q. Nguyen, "Model order reduction algorithm based on preserving dominant poles," *Int. J. Control. Autom. Syst.*, vol. 19, no. 6, pp. 2047–2058, 2021, doi: 10.1007/s12555-019-0990-8.

[36] A. C. Antoulas, *Approximation of Large-scale Dynamical Systems: An Overview*, vol. 37, no. 11. SIAM, 2004.

[37] A. K. Prajapati and R. Prasad, "Order reduction in linear dynamical systems by using improved balanced realization technique," *Circuits, Syst. Signal Process.*, vol. 38, no. 11, pp. 5289–5303, 2019, doi: 10.1007/s00034-019-01109-x.

[38] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN'95-international conference on neural networks*, vol. 4, pp. 1942–1948.

Vu Ngoc Kien received received a Ph.D. of control engineering and automation fromThai Nguyen University of Technology (TNUT), Vietnam, in 2015. He was appointed as an associate professor in 2022. He is currently the head of Electrical Equipment Department at the Faculty of Electrical Engineering. His research interests include modeling – simulation, robust control, model order reduced algorithm. https://orcid.org/0000-0002-2782-841X

Nguyen Tien Duy received received his MSc degree in information technology from Thainguyen University, Vietnam, in 2007. In 2017 he received PhD degree from Graduate University of Science and Technology, Vietnam Academy of Science and Technology, Vietnam, on Applied Mathematics. He is currently a lecturer at Thainguyen University of Technology, Thainguyen University, Vietnam. His research interests include Artificial Intelligence, Hedge-Algebras, Engineering controls, Computing. https://orcid.org/0000-0003-2636-071X

Dao Huy Du received a Ph.D of telecommunication engineering from HaNoi University of Science and Technology (HUST), Vietnam, in 2013. He was appointed as an associate professor in 2018. He is currently the head of Training Office. His research interests include Digital Processing, model order reduced algorithm, Mobile Communication. Optical fiber communications. https://orcid.org/0000-0002-9349-3814

Huy Nguyen Phuong received a Ph.D. of telecommunication engineering from HaNoi University of Science and Technology (HUST), Vietnam, in 2014. He is currently the head of Electronics Department at the Faculty of Electronics Engineering. His research interests include Digital Processing, Soft Computing, Optimal algorithm. https://orcid.org/0000-0002-5887-5796

Nguyen Hong Quang received a Ph.D. of control engineering and automation from Thai Nguyen University of Technology (TNUT), Vietnam, in 2019. He is currently a lecturer at the Faculty of Mechanical, Electrical, Electronics Technology. His research interests include electrical drive systems, control systems, and their applications, adaptive dynamic programming control, robust nonlinear model predictive control, motion control, and mechatronics. https://orcid.org/0000-0002-0433-3094