

Unmanned Bicycle Balance Control Based on Tunicate Swarm Algorithm Optimized BP Neural Network PID

Yun Li, College of Automation, Jiangsu University of Science and Technology, China*

 <https://orcid.org/0009-0003-0498-3630>

Yufei Wu, College of Mechanical Engineering, Jiangsu University of Science and Technology, China

Xiaohui Zhang, College of Automation, Jiangsu University of Science and Technology, China

Xinglin Tan, College of Mechanical Engineering, Jiangsu University of Science and Technology, China

Wei Zhou, College of Mechanical Engineering, Jiangsu University of Science and Technology, China

ABSTRACT

In this study, the authors introduce a novel approach that leverages the tunicate swarm algorithm (TSA) to optimize proportional-integral-derivative (PID) controller based on a back propagation (BP) neural network. The core objective of the approach is to manage and counteract uncertainties and disturbance that may jeopardize the balance and stability of self-driving bicycles in operation. By using the self-learning capabilities of BP neural networks, the controller can dynamically adjust PID parameters in real time. This enables an enhanced robustness and reliability during operation. Further bolstering the efficiency of our controller, the authors use the TSA to optimize the initial weights of a neural network. This effectively mitigates the commonly associated with slow convergence and being entrapped in local minima. Through simulation and experimentation, the findings reveal that the TSA-optimized BP neural network PID controller dramatically improves dynamic performance and robustness. It also proficiently manages changes in the environment such as wind and ground bumps. Therefore, the proposed controller design offers an effective solution to the balancing problem of self-driving bicycles and paves the way for a promising future in designing versatile controllers with broad application potential.

KEYWORDS

Back Propagation (BP) Neural Network, Proportional-Integral-Derivative (PID), Tunicate Swarm Algorithm (TSA), Unmanned Bicycles

INTRODUCTION

As a type of unmanned two-wheeled vehicles, unmanned bicycles have unique advantages, such as their agility and environmental friendliness. Therefore, they have great application prospects in various fields, such as avoiding traffic congestion, unmanned express delivery, and security patrols. These

DOI: 10.4018/IJITSA.324718

*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

prospects have attracted an increasing attention from experts and scholars for research on the motion balance control of unmanned bicycles (Su & Chen, 2020; Owczarkowski et al., 2019). Additionally, numerous solutions have been proposed to address the balance problem of unmanned bicycles, which was broadly classified into two categories: with and without stabilizers.

When no stabilizer is added to an unmanned bicycle, the most typical method used (Cui et al., 2020; Huang et al., 2017; Yongli et al., 2020; Huang et al., 2017) is to control the vehicle's turning angle. This is done by combining the rear wheel speed of the bicycle to generate centrifugal force to maintain its balance. Although this method does not require stabilizer addition, it can only maintain balance during motion and not during stillness.

The stabilizer problem of self-driving bicycles has various solutions. Some stabilizers use the principle of a gyroscope, which utilizes the energy produced by a rapidly rotating flywheel and then controls the angle of the flywheel to produce a force that helps maintain balance (Zheng et al., 2022; Park & Yi, 2020; Rózewicz & Piłat, 2020). This method can maintain a bicycle's balance when stationary and moving, and the flywheel response speed is fast, producing a large balance force. As suggested by CB et al. (2021), adding weight and adjusting the position on the bicycle can achieve balance by altering the bicycle's overall weight distribution. Although this method can also maintain a vehicle's balance when stationary and moving, adding weight increases its overall weight and size. The stabilizer used by Kim et al. (2015), Kien et al. (2023), Chiu and Wu (2020), Kim et al. (2013), and Kien et al. (2021) utilizes an inverted pendulum-based flywheel to generate a force that balances gravitational forces that act on the vehicle. This is achieved by modulating the flywheel's rotation, providing fast response times and instantaneous force. Furthermore, the flywheel's slower speed at the equilibrium state results in reduced power consumption.

Given the advantages and disadvantages of the various solutions mentioned, this study aims to establish a bicycle model that uses an inertial wheel utilizing the concept of an inverted pendulum.

Recent research on control algorithms for unmanned bicycles has been conducted. Zhang et al. (2023) proposed the Udwadia-Kalaba control method to balance unmanned bicycles by compensating for any deviation of the initial roll angle when the bicycle is stationary. This approach offers various benefits, such as rapid system response, low overshoot, and ease of control torque optimization. However, it only examined the stability of the bicycle when it was static and did not consider the balance problem when it moves forward. Li (2021) used cascaded PID control to integrate unmanned bicycle balance control and line-following control, resulting in enhanced control accuracy and faster system response. However, limited by the fixed PID parameters, PID control does not have a strong anti-interference ability when encountering disturbances during motion. Zhang et al. (2021) used the Kane method to establish an accurate mathematical model of unmanned bicycles and then proposed full-state feedback control that combines velocity correction and optimal control. They also used Linear Quadratic Regulator (LQR) to control a bicycle's balance. However, this method highly depends on an accurate mathematical model. During the bicycle's forward movement, it may encounter environmental factors (e.g., uneven road surfaces and wind instability interference). Moreover, the system is a typical nonlinear and underactuated control system (Zhang et al., 2020).

To address the shortcomings of prior studies and these challenges, we propose a novel method for balancing unmanned bicycles using a fine-tuned BP neural network PID controller using the tunicate swarm algorithm (TSA). The BP neural network PID does not depend on a spatial state model and can approximate complex nonlinear systems. The BP neural network PID controller can dynamically adjust and fine-tune itself to the system as the bicycle moves forward, enabling it to be highly robust and fault-tolerant. This allows the PID controller to handle external disturbances affecting the bicycle's balance effectively. However, the initial weight values of the BP neural network determine its convergence speed and the problem of getting stuck in local minima. Therefore, selecting appropriate initial weight values significantly impacts the performance of the entire PID controller. Typically, the network's initial weight values are randomly generated, which is highly uncertain and results in significant vibration and overshoot phenomena when the motor starts. Thus, the requirements of unmanned bicycle balance are difficult to meet. This study's innovation is the introduction of the TSA, which has a strong ability to

jump out of local optima and a high optimization capability (Sharma. et al., 2021), to globally optimize the initial weight values of a BP neural network. Then, fitness values are obtained by the output of the network's forward and feedback Kp, Ki, and Kd values, and the initial weight values are finally updated. TSA compensates for the disadvantages of the BP neural network, easily getting stuck in local minima and having a slow convergence speed (Wan & He, 2023), and enhances the dynamic performance of the BP_PID algorithm. This study uses simulation comparison to compare the performance and robustness of the BP neural network PID controller optimized using the TSA with three other controllers, highlighting the advantages of the proposed algorithm.

The remainder of this paper is arranged as follows. Section 2 provides a dynamic model for unmanned bicycles. The unmanned bicycle controller and optimization algorithms are presented in Section 3. Section 4 presents the conducted simulation comparisons and analyzes of different controller algorithms. Section 5 validates the optimized controller algorithm on the experimental platform of unmanned bicycles. Section 6 discusses the simulation and experimental results of this study and the limitations of the proposed method. Finally, Section 7 concludes the paper and provides suggestions for future research.

ESTABLISHMENT OF A DYNAMIC MODEL FOR UNMANNED BICYCLES

The Lagrange equation (Yetkin et al., 2014) is an energy-based approach that characterizes the motion equation of a system by considering the differences between a system's potential and kinetic energies. This method can transform a physical model into mathematical formulas, which can be used for the simulation and control design. Specifically, unmanned bicycles are complex systems with nonlinear and strong coupling characteristics. In this study, a momentum wheel, which can switch rapidly between forward and reverse directions, generates torque to counteract the gravitational torque applied to the bicycle in a semi-stable state, thus achieving balance. As the TSA only needs to optimize the initial weights of a BP neural network and during the initial startup phase of the bicycle, the bicycle is in a stationary state. The static transfer function derived from the static bicycle model greatly simplifies the unmanned bicycle model. The Lagrange equation is employed in this study to establish the dynamic model of an inertial wheel-based unmanned bicycle in a static state.

The Lagrange equation is derived from the universal equation of dynamics and represents the motion differential equations of a system of particles. The equation is expressed as:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = F_i, i = 1, 2, 3, \dots, n, \quad (1)$$

$$L = T - V. \quad (2)$$

Here, q_i , F_i , L , T , and V represent the generalized coordinates of the system, the generalized forces of the system, Lagrangian function, total kinetic energy, and total potential energy, respectively.

In Figure 2, θ represents the distance between the bicycle deviation and the vertical line of the ground, and ω WB denotes the rotation angle of the inertial wheel. A coordinate system is established using θ and ω as generalized coordinates. The model is also established as follows:

$$T_1 = \frac{1}{2} J_1 \dot{\theta}^2 \quad (3)$$

$$T_2 = \frac{1}{2} m_2 l_2^2 \dot{\theta}^2 + \frac{1}{2} J_2 (\dot{\theta} + \dot{\omega})^2 \quad (4)$$

Here, T_1 represents the total momentum of the bicycle frame; J_1 , moment of inertia of the bicycle frame about point O; T_2 , momentum of the flywheel; m_2 , mass of the inertial wheel; J_2 , moment

of inertia of the inertial wheel about its center; l_2 , the vertical distance from the center of mass of the inertial wheel GB to the ground; and l_1 , vertical distance from the center of gravity of the entire bicycle GA to the ground (Figure 1).

$$V = (m_1 l_1 + m_2 l_2) g \cos \theta \quad (5)$$

Here, V represents the total potential energy of the unmanned bicycle and m_1 is the mass of the bicycle frame.

Substituting the above equations into the Lagrange equation,

$$(J_1 + m_2 l_2 + J_2) \ddot{\theta} + J_2 \ddot{\omega} - (m_1 l_1 + m_2 l_2) g \sin \theta = 0, \quad (6)$$

$$J_2 (\ddot{\theta} + \ddot{\omega}) = -\tau \quad (7)$$

Here, A represents the force required to balance the unmanned bicycle.

The equation indicates the nonlinearity of the system. To simplify the controller design, the system is linearized at the operating point ($\theta=0$). Thus, $\sin \theta$ is approximated by θ . Consistent with the conventions of control theory, u represents the input torque - τ of the controlled object. The following differential equation can describe the resulting mathematical model of the system after linearization:

$$(J_1 + m_2 l_2 + J_2) \ddot{\theta} + J_2 \ddot{\omega} - (m_1 l_1 + m_2 l_2) g \theta = 0. \quad (8)$$

$$J_2 (\ddot{\theta} + \ddot{\omega}) = u. \quad (9)$$

Taking θ and u as the output and input in the equation, respectively, and applying Laplace transform, we obtain:

$$\frac{\theta(s)}{U(s)} = \frac{1}{(m_1 l_1 + m_2 l_2) g - (J_1 + m_2 l_2) s^2} \quad (10)$$

Table 1. Bicycle parameters

Name	Symbol	Value	Units
Inertial wheel mass	m_2	0.83	kg
Bicycle frame mass	m_1	14.53	kg
Vertical distance of the inertial wheel's center of mass to ground	l_2	0.61	m
Vertical distance from the vehicle center of mass to ground	l_1	0.53	m
Inertial wheel radius	r	0.90	m
Rotational inertia of the flywheel	J_2	0.30	$\text{g} \cdot \text{m}^2$
Rotational inertia of the bicycle body	J_1	0.17	$\text{kg} \cdot \text{m}^2$

Figure 1. Side view of a bicycle

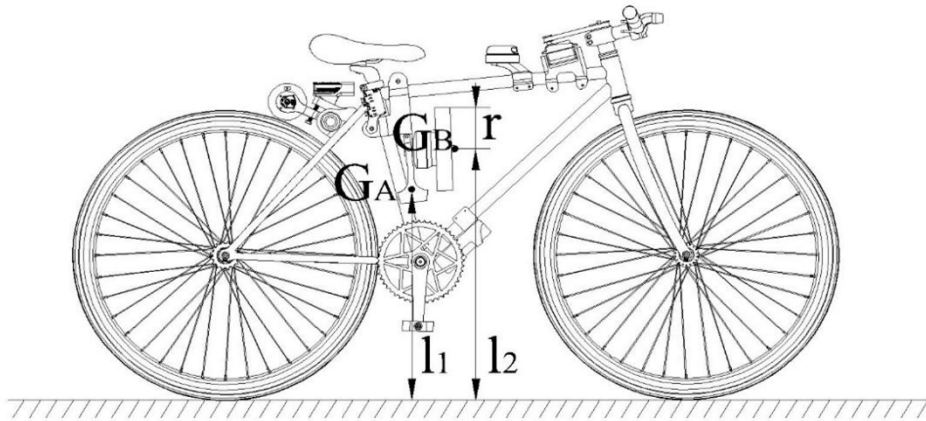
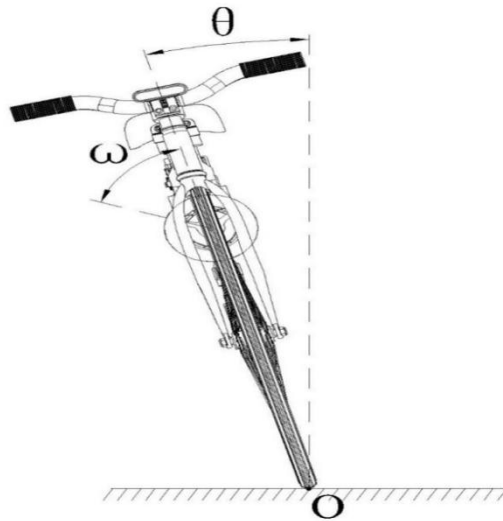


Figure 2. Front view of a tilted bicycle



THEORETICAL RESEARCH

TSA

The TSA is a novel optimization algorithm that draws inspiration from the propulsion and collective behavior of tunicates, sessile animals in the deep sea that navigate and forage through jet propulsion and group behavior (Kaur et al., 2020). This jet propulsion behavior primarily relies on an individual's gravity, the steady flow of interrupted water currents in the deep sea, and interaction forces between individuals to accomplish three tasks: avoiding conflicts between search agents, moving toward the position of the best search agent, and maintaining proximity to the best search agent. Conversely, the collective behavior is achieved through the tunicate's ability to perceive changes in water flow and the bioluminescence of their surrounding companions, allowing them to determine their positions and collectively converge toward the target food source. It can be generalized to update the position of all search agents regarding the current best solution.

To ensure that search agents do not overlap or collide, a new position for each search agent is calculated using $\vec{A}$. Here, $\vec{G}$, $\vec{F}$, and $\vec{M}$ represent the force of gravity, interrupted water currents in a steady flow, and interaction forces between individuals, respectively. The values of c_1 , c_2 , and c_3 are randomly generated within the range of $[0,1]$. P_{min} and P_{max} are typically set to 1 and 4 (Zhang et al., 2023), respectively. They represent the initial velocity of the interaction between individuals and the subordinate velocity, respectively. The following formula is used to calculate the new position:

$$\vec{A} = \frac{\vec{G}}{\vec{M}} \quad (11)$$

$$\vec{G} = c_2 + c_3 - \vec{F} \quad (12)$$

$$\vec{F} = 2 \cdot c_1 \quad (13)$$

$$\vec{M} = \left[P_{min} + c_1 \cdot (P_{max} - P_{min}) \right] \quad (14)$$

To move toward the position of the best search agent (represented by $\overrightarrow{FS}$), the following formula is used by the search agents. Here, $\overrightarrow{PD}$, $\overrightarrow{P_p}(x)$, and r_{and} represent the distance between the food source and search agent, position of the search agent at the current iteration, and a random number between 0 and 1, respectively.

$$\overrightarrow{PD} = \left| \overrightarrow{FS} - r_{and} \cdot \overrightarrow{P_p}(x) \right| \quad (15)$$

The search agents' orientation toward the direction of the optimal search agent is maintained. The formula for updating the position of the search individuals relative to the $\overrightarrow{FS}$ direction is expressed as $\overrightarrow{P_p}(x')$ and is expressed as follows:

$$\overrightarrow{P_p}(x') = \begin{cases} \overrightarrow{FS} + \vec{A} \cdot \overrightarrow{PD} & \text{if } r_{and} \geq 0.5 \\ \overrightarrow{FS} - \vec{A} \cdot \overrightarrow{PD} & \text{if } r_{and} < 0.5 \end{cases} \quad (16)$$

To simulate the collective behavior of swarming animals, the positions of other search agents are updated based on the positions of the top 2 optimal search agents. The following is the formula for updating the position of search agent $\vec{P_p}(x+1)$:

$$\vec{P_p}(x+1) = \frac{\vec{P_p}(x) + \vec{P_p}(x+1)}{2 + c_1} \quad (17)$$

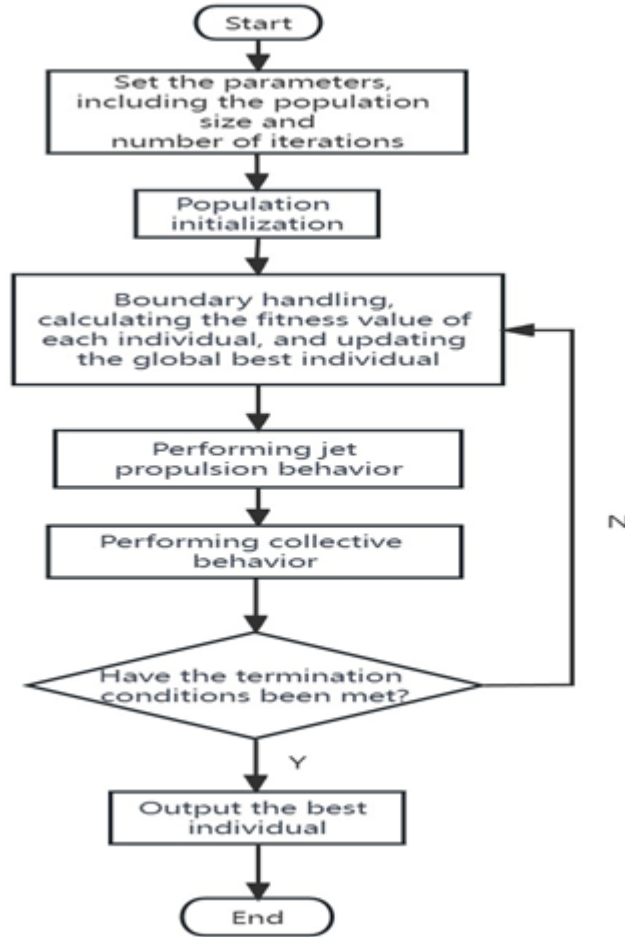
Figure 3 presents the flowchart of the TSA.

BP Neural Network PID

The incremental digital PID control algorithm can be expressed as follows:

$$u(k) = u(k-1) + \Delta u(k), \quad (18)$$

Figure 3. TSA flowchart



$$\Delta u(k) = K_p(e(k) - e(k-1)) + K_I e(k) + K_D(e(k) - 2e(k-1) + e(k-2)). \quad (19)$$

Here, $u(k)$, $u(k-1)$, $u(k)$, and $e(k)$ represent the output of the PID controller for the k th time, output of the $(k-1)$ th time, difference between the two outputs, and control error, respectively. Meanwhile, parameters K_p , K_I , and K_D in the algorithm are dynamically adjusted in real time through a neural network. The square of the output error $e(k)$ is used as the evaluation reference index $E(k)$.

$$E(k) = \frac{1}{2}[r(k) - y(k)]^2 = \frac{1}{2}e(k)^2 \quad (20)$$

The input of the network is

$$X = x(j) (j = 1, 2, 3). \quad (21)$$

One common activation function used in the hidden layer is the symmetrical Sigmoid function:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (22)$$

The output layer activation function is chosen as the positive Sigmoid function because K_p , K_i , and K_d must be positive.

$$g(x) = \frac{1}{1 + e^{-x}} \quad (23)$$

The output of the hidden layer is

$$L_1(k) = f\left(\sum_{j=0}^3 \omega_{ij} X_j(k)\right), \quad (24)$$

$$L_2^{(1)}(k) = K_p, L_2^{(2)}(k) = K_i, L_2^{(3)}(k) = K_d. \quad (25)$$

The output of the output layer is

$$L_2(k) = g\left(\sum_{k=0}^3 \omega_{jk} f\left(\sum_{j=0}^3 \omega_j X_j(k)\right)\right). \quad (26)$$

By using a gradient descent method, the formula for updating the weight values is

$$\Delta\omega(k) = -\eta \frac{\partial E}{\partial \omega} + \eta_1 \Delta\omega(k-1). \quad (27)$$

Thus,

$$\frac{\partial \Delta u(k)}{\partial L_2^{(1)}(k)} = e(k) - e(k-1), \quad (28)$$

$$\frac{\partial \Delta u(k)}{\partial L_2^{(2)}(k)} = e(k), \quad (29)$$

$$\frac{\partial \Delta u(k)}{\partial L_2^{(3)}(k)} = e(k) - 2e(k-1) + e(k-2). \quad (30)$$

TSA-Optimized BP_PID

The choice of initial weights in a BP neural network plays a crucial role in determining its convergence speed and tendency to get stuck in local minima (Zhang et al., 2022). Selecting appropriate initial weights is essential to ensure the optimal performance of the entire PID controller. Typically, a network's initial weights are randomly generated, leading to high uncertainty. To address this problem, the TSA performs global optimization on the initial weights of the BP neural network. The fitness value is obtained by feeding the output of K_p , K_i , and K_d values through the network's forward feedback. The initial weights ω are updated accordingly.

The unmanned bicycle controller employs cascaded PID for balance control. The angle loop utilizes TSA-optimized BP neural network PID, whereas the speed loop uses conventional PID to regulate the inertial wheel's speed. The angle of the unmanned bicycle's balance position serves as the target value, which is input into the angle loop. Consequently, the angle loop's output serves as the target value for the speed loop. Finally, the output of the speed loop is passed on to the inertial wheel, which counters the torque generated by bicycle tilting, thereby maintaining the bicycle's balance and stability. Figure 4 presents the block diagram illustrating this process.

COMPARISON OF THE SIMULATIONS OF THE DIFFERENT CONTROLLERS

This section presents a simulation comparison of TSA_BP_PID, PSO_BP_PID (Meng et al., 2022), unoptimized BP_PID (Ren et al., 2020), and conventional PID (Borase et al., 2021).

As presented in Figure 5 and Table 2, when subjected to a step input signal and no external disturbances, TSA_BP_PID and PSO_BP_PID controllers exhibit shorter rise times than BP_PID and PID controllers. Furthermore, the TSA_BP_PID controller has the shortest adjustment time, whereas the BP_PID controller has the smallest overshoot. As a result, the overshoot of TSA_BP_PID is larger than that of BP_PID. However, considering the comprehensive factors of adjustment time, rise time, and overshoot, the dynamic performance of the TSA_BP_PID controller is better than that of the BP_PID controller, achieving the optimization effect. Moreover, the dynamic performance of TSA_BP_PID is slightly better than that of PSO_BP_PID.

Figure 6 indicates that, when subjected to an impulse disturbance at 1.5 s, the TSA_BP_PID controller achieved convergence within 0.2 s, which is similar to the convergence speed of the PSO_BP_PID controller, but much faster than the other two controllers. In addition, the PSO_BP_PID, BP_PID, and PID controllers all exhibited some overshoot, but the TSA_BP_PID controller did not. Overall, these results demonstrate that the TSA_BP_PID controller exhibits superior robustness.

Figure 4. Block diagram of the control system

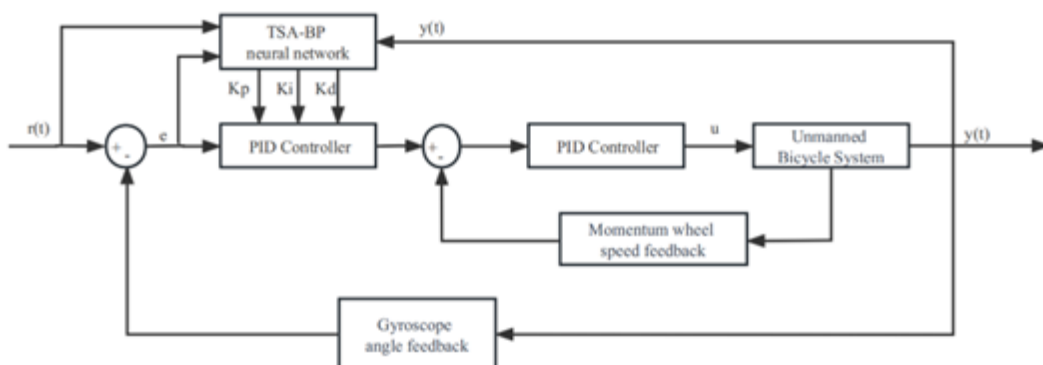


Figure 5. Simulation comparison of the control algorithms without disturbances

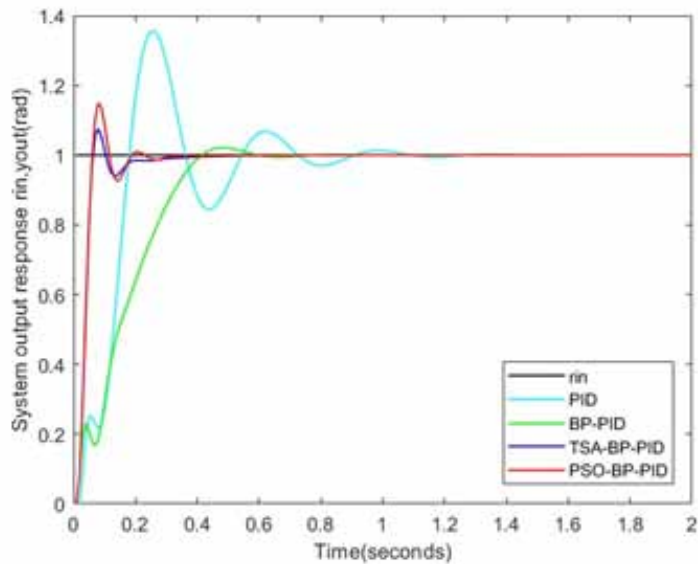


Table 2. Comparison of control technique performance

Control Technique	Rise Time (s)	Settling Time (s)	Overshoot
PID	0.14	0.36	35%
BP-PID	0.31	0.49	2%
TSA-BP-PID	0.04	0.11	7%
PSO-BP-PID	0.04	0.12	14%

Figure 6. Simulation comparison of the control algorithms under disturbance

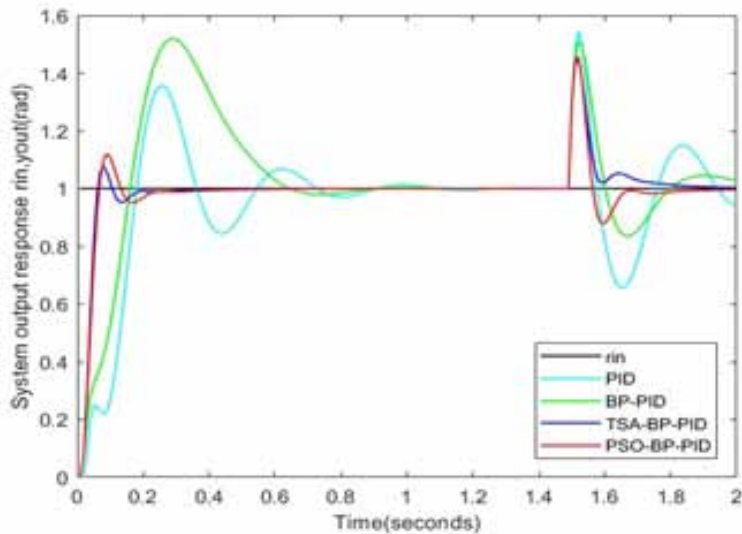


Figure 7 shows the comparison of fitness curves between PSO_BP_PID and TSA_BP_PID. The results show that the PSO algorithm approaches the minimum fitness value of about 0.0137 at around three iterations. Meanwhile, the TSA algorithm approaches the minimum fitness value of about 0.013 at around 27 iterations. Figure 8 displays the average execution time for each algorithm, with PSO taking 14.965701 seconds for 100 iterations and TSA taking 5.960677 seconds for 100 iterations. Figure 9 shows the optimized values of K_p , K_i , and K_d through PSO and TSA, with $K_p=17.5$, $K_i=0.18$, and $K_d=9.4$ after PSO optimization, and $K_p=12.1$, $K_i=0.67$, and $K_d=3.6$ after TSA optimization. We can conclude that although the TSA algorithm requires more iterations to achieve the minimum fitness value, it runs faster than the PSO algorithm.

Figure 7. Fitness curve

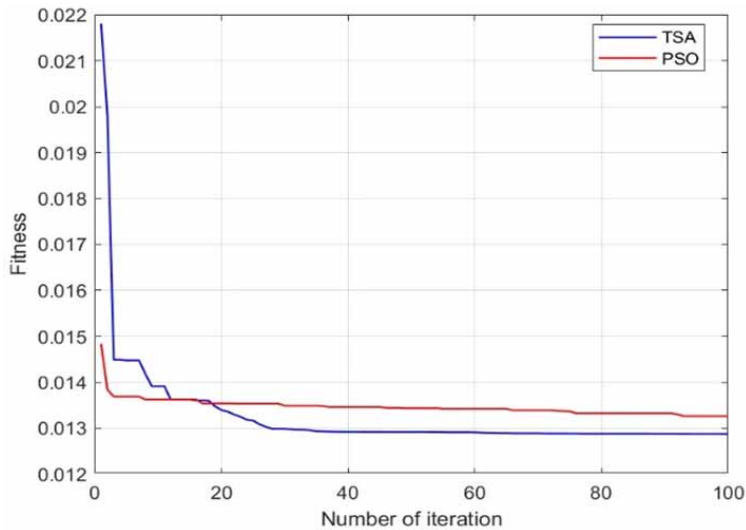


Figure 8. Comparison of the execution time

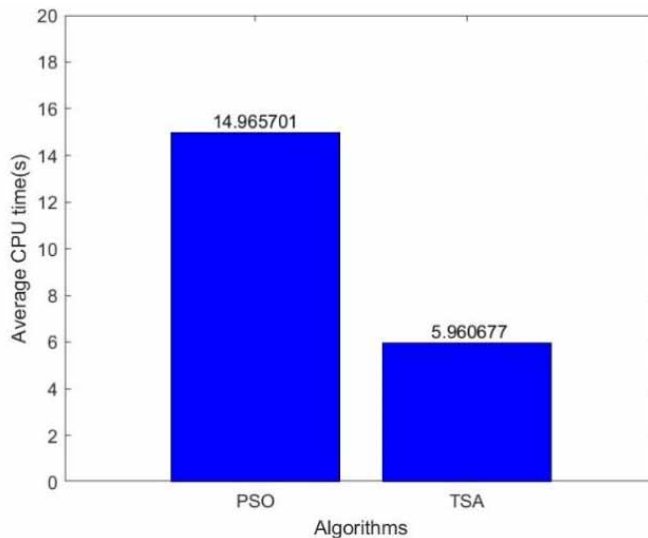
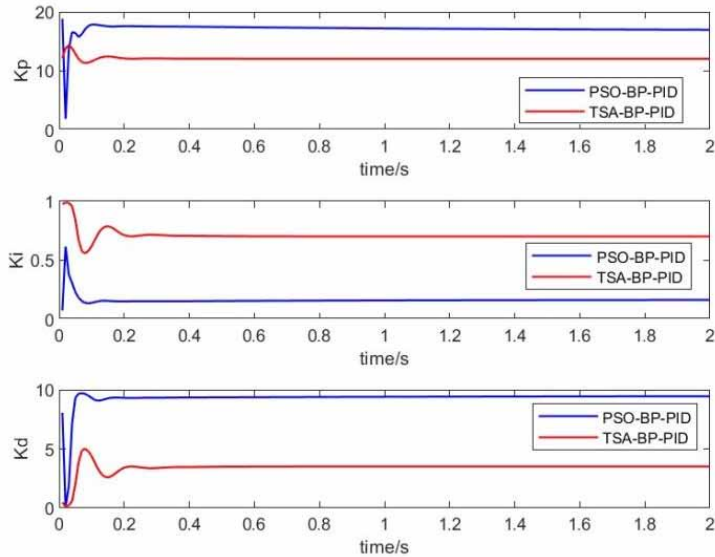


Figure 9. K_p , K_i , and K_d after optimization



VALIDATION OF UNMANNED BICYCLE BALANCE

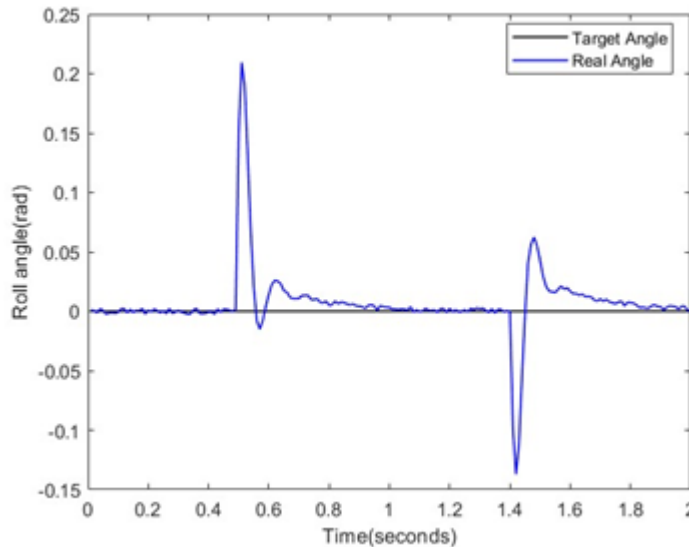
An experimental platform for an unmanned bicycle was constructed (Figure 10) to demonstrate the feasibility of the TSA_BP_PID control algorithm. The platform utilized Intel NUC12WSKi500 as the control component and was equipped with a ROS operating system based on Ubuntu 18.04. The TSA-BP-PID control algorithm was applied to send signals via a USB-CAN module to an Odrive driver board, which controlled the rotation of the inertial wheel motor to generate a force that countered the gravitational force acting on the bicycle.

The TSA-BP-PID controller was used to maintain the balance of an unmanned bicycle on an uneven surface while simulating the possible uncertainties, such as wind disturbances through object impacts during actual movements. Figure 11 compares the measured actual roll angle of the bicycle using the

Figure 10. Experimental platform of the unmanned bicycle



Figure 11. Roll angle during bicycle movement



gyroscope and the target roll angle to demonstrate the system's performance. The TSA-BP-PID controller quickly restored balance in response to sudden impact disturbances, demonstrating its robustness.

DISCUSSION

In this study, we investigated the performance of a BP neural network PID controller optimized using the TSA in the simulation comparison and experimental platform validation. Our results indicate that the TSA-optimized BP neural network PID controller exhibited better robustness and dynamic performance than the unoptimized BP neural network PID controller and had smaller overshoot and settling time than the BP neural network PID controller optimized by using a particle swarm optimization (PSO) algorithm. This outcome is because the PSO algorithm has poor exploration ability in the search space, leading to the problem of being trapped in locally optimal solutions. Conversely, the search mechanism of the TSA provides a good balance between exploration and exploitation (Sharma et al., 2021). This results in better optimization performance of the BP neural network PID controller.

CONCLUSION

This study focused on the design of a balance controller for unmanned bicycles. First, we developed a mathematical model of an unmanned bicycle that we then analyzed based on a momentum wheel so we could derive a transfer function. A new type of controller that integrates the TSA with the BP neural network PID control was proposed to optimize the initial weights of a BP neural network. In simulation comparisons, the TSA-BP-PID controller exhibited superior dynamic performance and robustness compared to the other three controllers. Finally, we undertook experimental investigations with a physical prototype to demonstrate how the controller can quickly restore balance in uneven road surfaces and sudden impact disturbances. Uncertainty perturbations caused by environmental factors affecting unmanned bicycles' balance and stability were addressed.

However, our proposed controller remains to have certain limitations. For example, we only optimized the initial weights of the BP neural network using the TSA algorithm. However, the BP neural network can also be optimized by adjusting the learning rate, increasing the network depth, and using regularization. These could be directions for future research. Future work could explore integrating SLAM navigation technology with unmanned bicycles to enable autonomous navigation while maintaining balance and stability across various terrains. In addition, this investigation could enhance research in practical applications, such as unmanned delivery and security patrols.

AUTHOR NOTE

The data used to support the findings of this study are provided in this paper.

The author declares that there is no conflict of interest regarding the publication of this paper.

This research received no external funding.

REFERENCES

- Borase, R. P., Maghade, D. K., Sondkar, S. Y., & Pawar, S. N. (2021). A review of PID control, tuning methods and applications. *International Journal of Dynamics and Control*, 9(2), 818–827. doi:10.1007/s40435-020-00665-4
- C B, A., praseeth, D., Babu, J., Ryan, M., & G, T. (2021). Gyroscopic hybrid bicycle. *International Research Journal on Advanced Science Hub*, 3(1S), 20–26. doi:10.47392/irjash.2021.014
- Chiu, C. H., & Wu, C. Y. (2020). Bicycle robot balance control based on a robust intelligent controller. *IEEE Access : Practical Innovations, Open Solutions*, 8, 84837–84849. doi:10.1109/ACCESS.2020.2992792
- Cui, L., Wang, S., Lai, J., Chen, X., Yang, S., Zhang, Z., & Jiang, Z. P. (2020, October). Nonlinear balance control of an unmanned bicycle: Design and experiments. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7279–7284. doi:10.1109/IROS45743.2020.9341150
- Huang, C. F., Tung, Y. C., & Yeh, T. J. (2017). Balancing control of a robot bicycle with uncertain center of gravity. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 5858–5863. doi:10.1109/ICRA.2017.7989689
- Huang, C. F., Tung, Y. C., & Yeh, T. J. (2017). Balancing control of a robot bicycle with uncertain center of gravity. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 5858–5863. doi:10.1109/ICRA.2017.7989689
- Kaur, S., Awasthi, L. K., Sangal, A. L., & Dhiman, G. (2020). Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Engineering Applications of Artificial Intelligence*, 90, 103541. doi:10.1016/j.engappai.2020.103541
- Kien, V. N., Duy, N. T., Du, D. H., Huy, N. P., & Quang, N. H. (2023). Robust optimal controller for two-wheel self-balancing vehicles using particle swarm optimization. *International Journal of Mechanical Engineering and Robotics Research*, 12(1), 16–22. doi:10.18178/ijmerr.12.1.16-22
- Kien, V. N., Quang, N. H., & Trung, N. K. (2021). Application of model reduction for robust control of self-balancing two-wheeled bicycle. *TELKOMNIKA Telecommunication Computing Electronics and Control*, 19(1), 252–264. doi:10.12928/telkomnika.v19i1.16298
- Kim, H. W., An, J. W., Dong Yoo, H., & Lee, J. M. (2013). Balancing control of bicycle robot using PID control. *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, 145–147. doi:10.1109/ICCAS.2013.6703879
- Kim, Y., Kim, H., & Lee, J. (2015). Stable control of the bicycle robot on a curved path by using a reaction wheel. *Journal of Mechanical Science and Technology*, 29(5), 2219–2226. doi:10.1007/s12206-015-0442-1
- Li, F. (2021). Research on self-balancing unmanned bicycle based on cascade PID control. *2021 IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, 442–446. doi:10.1109/ICIBA52610.2021.9687999
- Meng, Z., Zhang, L., Wang, H., Ma, X., Li, H., & Zhu, F. (2022). Research and design of precision fertilizer application control system based on PSO-BP-PID algorithm. *Agriculture*, 12(9), 1395. doi:10.3390/agriculture12091395
- Owczarkowski, A., Horla, D., & Zietkiewicz, J. (2019). Introduction of feedback linearization to robust LQR and LQI control—analysis of results from an unmanned bicycle robot with reaction wheel. *Asian Journal of Control*, 21(2), 1028–1040. doi:10.1002/asjc.1773
- Park, S. H., & Yi, S. Y. (2020). Active balancing control for unmanned bicycle using scissored-pair control moment gyroscope. *International Journal of Control, Automation, and Systems*, 18(1), 217–224. doi:10.1007/s12555-018-0749-7
- Ren, H., Hou, B., Zhou, G., Shen, L., Wei, C., & Li, Q. (2020). Variable pitch active disturbance rejection control of wind turbines based on BP neural network PID. *IEEE Access : Practical Innovations, Open Solutions*, 8, 71782–71797. doi:10.1109/ACCESS.2020.2987912

- Różewicz, M., & Piłat, A. (2020). Robust controller based on Kharitonov Theorem for bicycle with CMG. In A. Bartoszewicz, J. Kabziński, & J. Kacprzyk (Eds.), *Advanced, contemporary control. advances in intelligent systems and computing* (Vol. 1196, pp. 411–423). Springer International Publishing. doi:10.1007/978-3-030-50936-1_35
- Sharma, A., Dasgotra, A., Tiwari, S. K., Sharma, A., Jatly, V., & Azzopardi, B. (2021). Parameter extraction of photovoltaic module using tunicate swarm algorithm. *Electronics (Basel)*, 10(8), 878. doi:10.3390/electronics10080878
- Su, S., & Chen, G. (2020). Lateral robust iterative learning control for unmanned driving robot vehicle. *Proceedings of the Institution of Mechanical Engineers. Part I, Journal of Systems and Control Engineering*, 234(7), 792–808. doi:10.1177/0959651820904834
- Wan, H.-C., & He, Y. (2023). PID control of spinning frame based on improved PSO-BP neural network. *Manufacturing Automation*, 45(2), 159–163.
- Yetkin, H., Kalouche, S., Vernier, M., Colvin, G., Redmill, K., & Oztuner, U. (2014). Gyroscopic stabilization of an unmanned bicycle. *2014 American Control Conference*, 4549–4554. doi:10.1109/ACC.2014.6859392
- Yongli, Z., Liu, Y., & Yi, G. (2020). Model analysis of unmanned bicycle and variable gain lqr control. *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, 1196–1200. doi:10.1109/ITAIC49862.2020.9338891
- Zhang, G., Sun, B., Bai, W., & Zhang, H. (2022). Prediction of the yield performance and failure mode of RC columns under cyclic-load by PSO-BP neural network. *Buildings*, 12(5), 507. doi:10.3390/buildings12050507
- Zhang, J.-L., Zhao, R.-Y., Feng, Y.-L., Yang, H., & Wu, L.-L. (2023). A balance control method for bicycle robots based on Udwadia–Kalaba theory. *Chinese Journal of Engineering*, 45(2), 318–325. doi:10.13374/j.issn2095-9389.2021.08.27.007
- Zhang, Y., Liu, Y., & Yi, G. (2020). Model analysis of unmanned bicycle and ADRC control. *2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, 953–957. doi:10.1109/ICIBA50161.2020.9277227
- Zhang, Y., Zhao, G., & Li, H. (2021). Multibody dynamic modeling and controlling for unmanned bicycle system. *ISA Transactions*, 118, 174–188. doi:10.1016/j.isatra.2021.02.014 PMID:33632599
- Zheng, X., Zhu, X., Chen, Z., Sun, Y., Liang, B., & Wang, T. (2022). Dynamic modeling of an unmanned motorcycle and combined balance control with both steering and double CMGs. *Mechanism and Machine Theory*, 169, 104643. doi:10.1016/j.mechmachtheory.2021.104643