Adaptive control of ball and beam system using SNA-PID combined with recurrent fuzzy neural network identifier

Minh-Thanh Le[1], Chi-Ngon Nguyen[2]
[1]Department of Control and Automation Engineering, Faculty of Electrical and Electronics, Vinh Long University of Technology Education, Vinh Long, Vietnam
[2]Faculty of Automation Engineering, College of Engineering, Can Tho University, Can Tho, Vietnam

Article Info

Article history:

Received Jun 12, 2025
Revised Jan 19, 2026
Accepted Jan 25, 2026

Keywords:

Ball and beam system
Intelligent control
MATLAB/Simulink
Nonlinear system
Proportional integral derivative
SNA-PID-RFNNI

ABSTRACT

The ball and beam system is a nonlinear and inherently unstable single-input, multiple-output (SIMO) system, which poses significant challenges for control design. Intelligent control algorithms are often applied to autonomously control complex systems when there are changes in parameters or the control environment. Therefore, in this paper, we research and develop two methods: proportional integral derivative (PID) and single neuron adaptive (SNA)-PID-recurrent fuzzy neural network identifier (RFNNI) to control the ball and beam system. Simulation results on MATLAB/Simulink show that the SNA-PID-RFNNI controller provides a more stable output signal than the traditional PID controller, with minimal overshoot and a settling time of about 15 seconds. Next, we will conduct real-time experiments on the object using the proposed algorithm through the MEGA2560 control board with an ultrasonic positioning mechanism.

Corresponding Author:

Minh-Thanh Le
Department of Control and Automation Engineering, Faculty of Electrical and Electronics
Vinh Long University of Technology Education
Vinh Long, Vietnam
Email: thanhlm@vlute.edu.vn

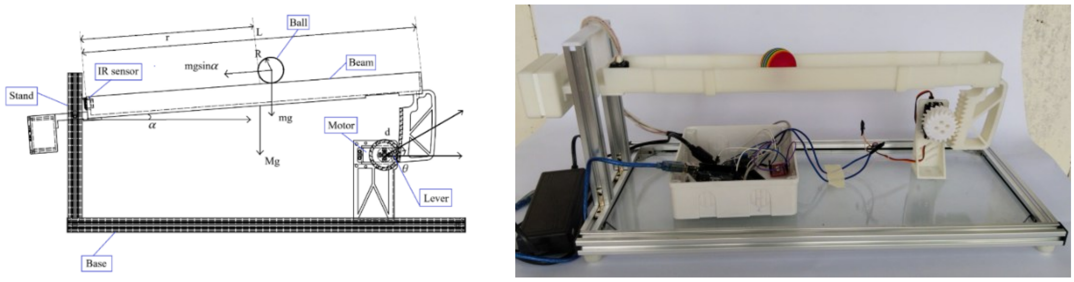
1. INTRODUCTION

The ball and beam system is a well-known classical benchmark model in the fields of automatic control research and education, due to its unstable and nonlinear characteristics [1]. In terms of physical structure, the system consists of a ball capable of rolling along a long rod. This rod has one fixed end; the other end can be tilted by controlling a motor with a connecting rod attached to it; the control goal is to keep the ball stable at a pre-defined location on the rod [2]. Due to the influence of gravity, the material of the ball and the slider, when adjusting the tilt angle of the rod (even very small), the ball will roll quickly to one side, causing the system to fall into an unbalanced state easily [3]. The unstable and nonlinear characteristics make the ball-and-rod balancing system ideal for testing modern control algorithms, from classical proportional integral derivative (PID) control to intelligent methods such as fuzzy control, neural networks, and hybrid methods [4], [5]. PID control is still the most popular solution due to its ease of tuning and significant efficiency in practice. However, classical PID controllers are often designed with fixed parameters, leading to poor performance when the control object changes or when there is significant noise and signal delays [6].

To improve the adaptability and accuracy of the system, a prominent approach is to use an adaptive single neuron network to optimize the PID controller, where the coefficients Kp, Ki, and Kd are adjusted online through the neural network [7]. In addition, accommodating the nonlinear and time-dependent characteristics of rod and ball system, the system identifier plays an essential role in providing information for adaptive control. Recurrent fuzzy neural network identifier (RFNNI) is a suitable tool for modeling and identifying nonlinear systems that can handle time and fuzzy inference [8]. This paper proposes a new control method combining a single-neuron adaptive PID controller and RFNNI, applied to a ball and beam system. This method is verified through simulation on the MATLAB/Simulink platform, showing that the algorithms help improve accuracy, reduce the transient time, and improve the system's stability [9], [10].

2. MATHEMATICAL MODELLING

The beam and ball system consists of a circular ball that can roll along a long beam. A motor placed at one end of the beam adjusts the angle of inclination of the beam, and a connecting rod connects the rotating disk to the beam [9]. The system is depicted in Figure 1, where Figure 1(a) shows the system modeling and Figure 1(b) shows the experimental prototype. The system parameters are presented in Table 1.



(a) (b)

Figure 1. Modeling and experimental setup of the ball and beam system: (a) system modeling and (b) experimental prototype

Table 1. Actual parameters on the model that the team has built

Parameter	Unit	Value	Meaning
L	*(m)*	0.4	Length of the beam
m	*(kg)*	0.11	Mass of the ball
M	*(kg)*	0.4	Mass of the beam
g	*(m/s2)*	9.8	Gravitational acceleration
r	*(m)*	0.2	Ball position
R	*(m)*	0.015	Radius of the ball
α	*(rad)*	0	Beam angle coordinate
θ	*(rad)*	0	Servo gear angle
d	*(m)*	0.02	Lever arm offset

The ball dynamics are primarily influenced by gravity and its moment of inertia. This results in a system that is unstable and nonlinear. The development of an effective controller requires an accurate mathematical representation of the system [1].

According to the literature [10]–[13], the mathematical equation of plant is described as (1) and (2).

$$(mr^2 + J_M)\ddot{\alpha} + 2mr\dot{r}\dot{\alpha} + \frac{L}{2}gM\cos\alpha + gmr\cos\alpha = \tau \tag{1}$$

$$\left(m + \frac{J_m}{R^2}\right)\ddot{r} - mr\dot{\alpha}^2 + mg\sin\alpha = 0 \tag{2}$$

With J_M, J_m are the moments of inertia of the beam and ball, τ is the torque generated by the motor. Let $x_1 = r, x_2 = \dot{r}, x_3 = \alpha, x_4 = \dot{\alpha}$ [9], we get (3) and (4).

$$\dot{x}_1 = x_2; \dot{x}_2 = \frac{mx_1x_4^2 - mg\sin x_3}{m + \frac{Jm}{R^2}} \tag{3}$$

$$\dot{x}_3 = x_4; \dot{x}_4 = \frac{\tau - 2mx_1x_2x_4 - \frac{L}{2}gM\cos x_3 - gmx_1\cos x_3}{mx_1^2 + J_M} \tag{4}$$

This model describes the interaction between the beam angle and the ball position. Based on (3) and (4), the system is implemented in MATLAB/Simulink environment, as illustrated in Figure 2, forming the basis for the subsequent controller design.

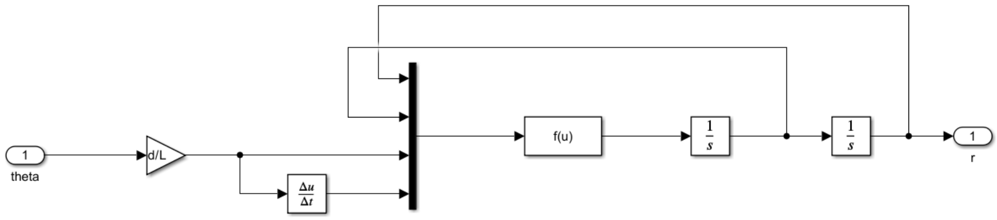


Figure 2. Modeling of ball and beam system in MATLAB/Simulink

3. DESIGN OF THE CONTROLLER

In this study, the controller is proposed based on the single neuron adaptive (SNA)-PID algorithm. It is combined with the RFNNI to control the ball and beam system. The goal is to adapt and respond quickly to changes in the system state while handling the nonlinear and noisy characteristics of the entire closed-loop system well.

3.1. The single neuron adaptive-proportional integral derivative controller

Figure 3 depicts the structure of PID controller. The controller is constructed using a linear neuron. This design integrates proportional, integral, and derivative components within a neural-based framework.

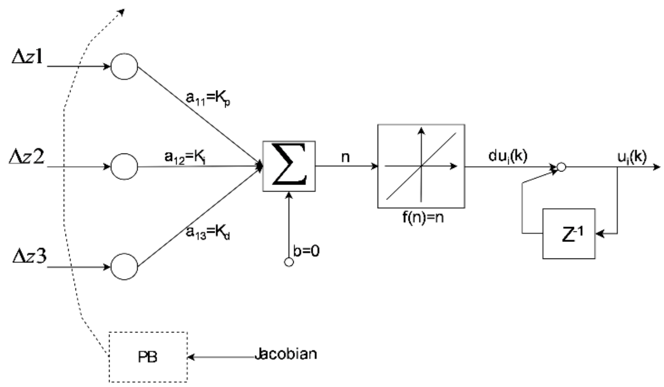


Figure 3. Structure of SNA-PID controller

As given in (5), the proportional, integral, and derivative terms are computed from the error between the reference input and the system output.

$$\Delta z1 = z(k); \Delta z2 = \int_0^{\infty} z(k)dk; \Delta z3 = \frac{dz(k)}{dk} \tag{5}$$

The equation describing the PID controller is defined as (6) [14], [15].

$$u(k) = K_p \Delta z1 + K_i \Delta z2 + K_d \Delta z3 + u(k-1) \tag{6}$$

Here, *z(k)* (with *k* =1, 2, 3) denotes the error between the reference signal and the system response. In the proposed controller, the neuron output is equivalently regarded as the PID controller output, which is shown in (7) [16].

$$n = (a_{11}\,\Delta\, z1 + a_{12}\Delta z2 + a_{13}\Delta z3)$$
$$du(k) = f(n) = n \; and \; u(k) = du(k) + u(k-1) \tag{7}$$

Where, $a_{1i|I=1,2,3}$ are the weights of the neurons, corresponding to the PID gain set (K_p, K_i, K_d), and they are adaptively updated in real time.

The training process for the SNA-PID controller aims to minimize the cost function shown in (8) by updating the network weights $a_{1i|i=1,2,3}$.

$$Z(k) = \frac{1}{2} z^2(k) \; with \; z(k) = y_{ref}(k) - y(k) \tag{8}$$

Here, $y_{ref}(k)$ and *y(k)* denote reference signal and system response, respectively. The gradient descent method was employed to adapt the network weight set $a_{1i|i=1,2,3}$, as described in [17], [18].

3.2. The recurrent fuzzy neural network identifier

The RFNNI as shown in Figure 4, is a multi-layer recurrent neural network for fuzzy inference that can be formulated using a set of fuzzy inference rules [17].

i) Layer 1 - input layer: the RFNNI uses the current control input and the previous system output as input variables, whose connection weights are updated at the current time instant k as expressed in (9). Specifically, the input signals are defined as (10).

$$G_i^1(k) = x_i^1(k) + \theta_i^1 G_i^k(k-1)\ with\ i = 1,2 \tag{9}$$

$$x_1^1(k) = u(k)\ and\ x_2^1(k) = y(k-1) \tag{10}$$

ii) Layer 2 - fuzzy layer: this layer is composed of (2×5) nodes, each node represents a Gaussian function with mean m_{ij}, standard deviation σ_{ij}, as defined by (11).

$$G_{ij}^2(k) = exp\left[-\frac{\left(G_i^1(k)-m_{ij}\right)^2}{\sigma_{ij}}\right], with\ i = 1,2\ and\ j = 1,2,3,4,5 \tag{11}$$

The RFNNI's online learning mechanism involves the adaptive adjustment of two specific parameters m_{ij} and σ_{ij} for each node within the fuzzy layer.

iii) Layer 3 - rule layer: this layer contains (5×5) nodes, and the output of the node q is given by (12).

$$G_q^3(k) = \prod_i G_{iq_i}^2(k)\ with\ i, q_i = 1,2,3,4,5 \tag{12}$$

iv) Layer 4 - output layer: it consists of a linear neuron, and its output is expressed as (13).

$$G_i^4(k) = \sum_j a_{ij}^4 G_j^3(k)\ with\ i = 1\ and\ j = 1,2,\ldots,25 \tag{13}$$

With a_{ij}^4 is the connection weight between layer 3 and layer 4. The output of this layer also represents the output of the RFNNI, as expressed in (14).

$$G_1^4(k) = y_m(k) = \widehat{f}\ [x_1(k), x_2(k)] with\ x_1(k), x_2(k)\ are\ defined\ by\ (10) \tag{14}$$

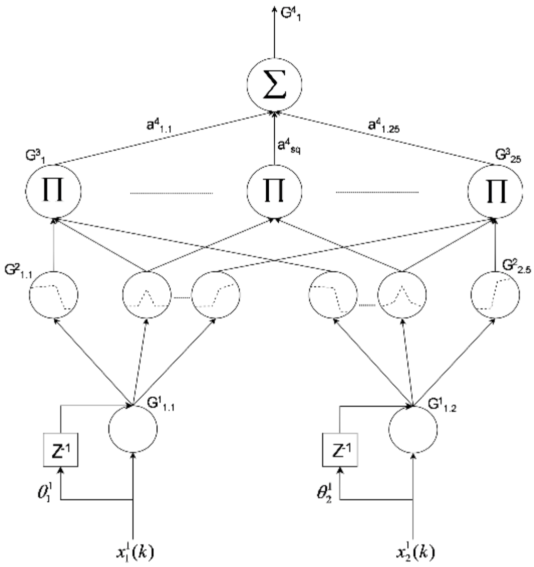


Figure 4. The structure of the RFNNI

The purpose of the online training algorithm in RFNNI is to adaptively tune the network weights and the fuzzy layer-dependent function parameters to minimize the cost function, as defined in (15).

$$Z(k) = \frac{1}{2}[y(k) - y_m(k)]^2 = \frac{1}{2}[y(k) - G_1^4(k)]^2 \tag{15}$$

Using the backpropagation technique, the RFNNI connection weight set will be adjusted in (16).

$$A(k+1) = A(k) + \Delta A(k) \; with \; \Delta A(k) = \eta\left(-\frac{\partial Z(k)}{\partial A}\right) \tag{16}$$

Where, η and A denote the learning rate constant and the adjustable parameter in the RFNNI training, respectively. The gradient of *Z(.)* in (15) and the weight of each RFNNI network layer are determined in reference [17]. Additionally, to predict the output $y_m(k)$ of the plant model, the RFNNI must also compute the Jacobian $\frac{\partial y(k)}{\partial u(k)}$, which is essential for the online training of the SNA-PID controller. The Jacobian information is defined in (17), referring to [17], [19].

$$\frac{\partial y(k)}{\partial u(k)} = \frac{\partial G_1^4}{\partial u} = \sum_{q=1}^{25}\left\{\frac{\partial G^4}{\partial G_q^3}\cdot\frac{\partial G_q^3}{\partial u}\right\} = \sum_{q=1}^{25} A_{ij}^4 \cdot \left\{\frac{\partial G_q^3}{\partial u}\right\} = \sum_q a_{ij}^4 \cdot \left\{\sum_s \frac{\partial G_q^3}{\partial G_{qs}^2}\cdot\frac{\partial G_{qs}^2}{\partial u}\right\} = \sum_q A_{ij}^4 \cdot \left\{\sum_s \frac{\partial G_q^3}{\partial G_{qs}^2}\cdot\frac{(-2)\left[G_{ij}^1(k) - m_{ij}\right]}{\left(\sigma_{ij}\right)^2}\right\} \tag{17}$$

4. SIMULATION RESULTS AND DISCUSSION

4.1. Diagram of SNA–PID–RFNNI controller

A block diagram of the control system developed using MATLAB/Simulink is presented in Figure 5 [20], [21]. Based on (3) and (4), the mathematical description for the ball and beam setup is established. The parameters of the SNA-PID controller are selected by a trial-and-error method, as described in [4]. The structure of the RFNNI set consists of 4 main layers: input, fuzzification, rule, and defuzzification. The recognition set supports online adjustment through backpropagation gradient. It outputs the output signal of the object as well as the Jacobian derivative to determine the influence level of the input on the output, serving the training of adaptive control.

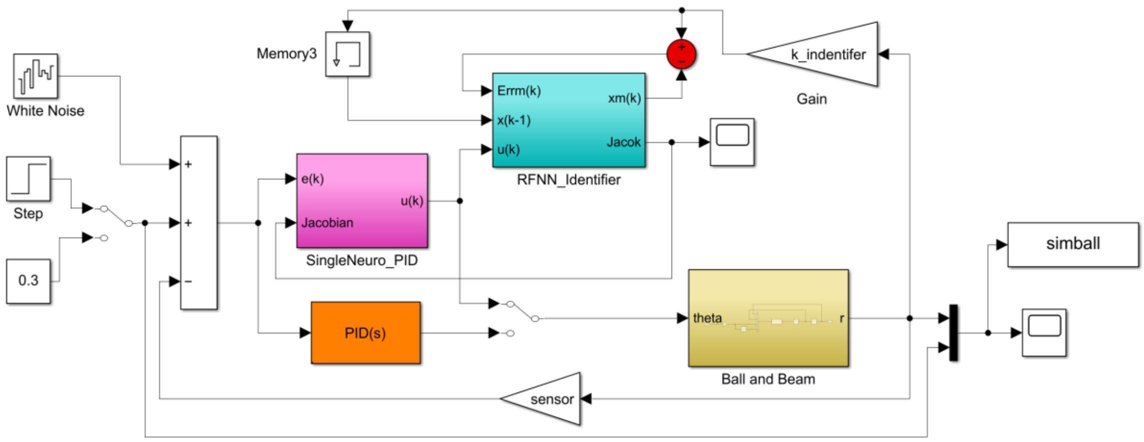


Figure 5. SNA-PID-RFNNI controller diagram [19], [22]

4.2. Results and comparison

The beam and ball system, using the SNA-PID-RFNNI algorithm, is simulated with fixed parameter positions of 24 cm and 28 cm (step function) and zero noise for the output responses in Figures 6 and 7. The results show that the ball moves and stays at the desired position. Continue using the SNA-PID-RFNNI algorithm to simulate with fixed parameter positions of 30 cm and 35 cm (constant function) with a noise of 0.00001 for the output responses in Figures 8 and 9. The results show that the ball still maintains its desired position, although there is a slight oscillation around the equilibrium point.

The beam and ball system using the PID and the SNA-PID-RFNNI algorithm is simulated with fixed parameter positions of 26 cm (step function) with zero noise for the output responses in Figure 10, with fixed parameter positions of 32 cm (constant function) with noise equal to 0.00001 for the output responses in Figure 11. In Figure 10, both the SNA-PID-RFNNI and PID controllers return the ball to the reference position. However, the PID controller causes large oscillations and higher initial overshoot, while the SNA-PID-RFNNI controller provides a smoother response, shorter recovery time, and less oscillation. When disturbances are present (Figure 11), the difference becomes more pronounced. The PID controller continues to exhibit large overshoots and prolonged settling times. In contrast, the SNA-PID-RFNNI maintains a response close to the reference position with fewer oscillations and improved noise immunity.

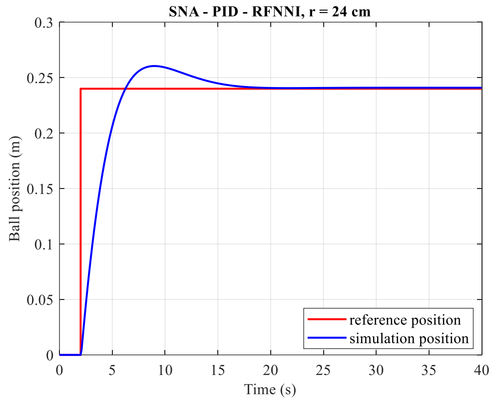


Figure 6. Transient response of SNA-PID-RFNNI with r =24 cm

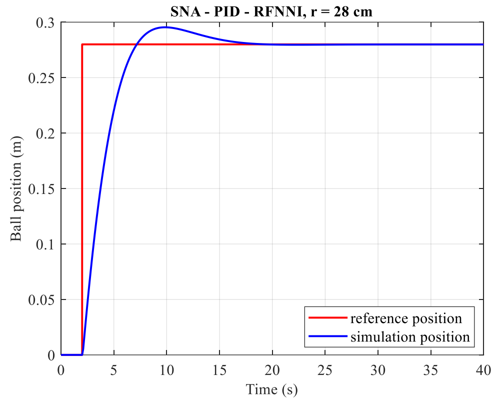


Figure 7. Transient response of SNA-PID-RFNNI with r =28 cm

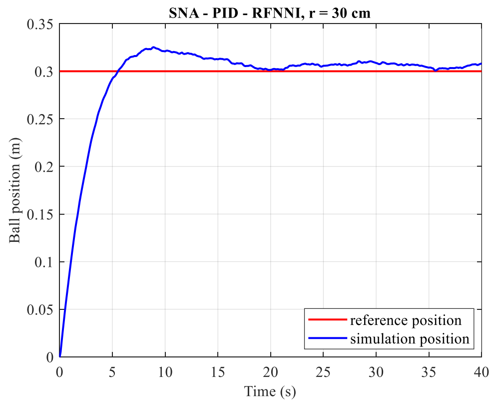


Figure 8. Transient response of SNA-PID-RFNNI with r =30 cm

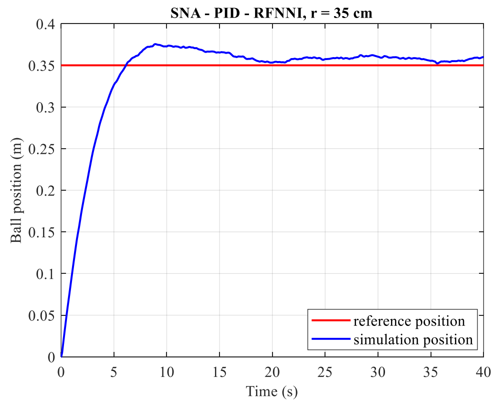


Figure 9. Transient response of SNA-PID-RFNNI with r =35 cm

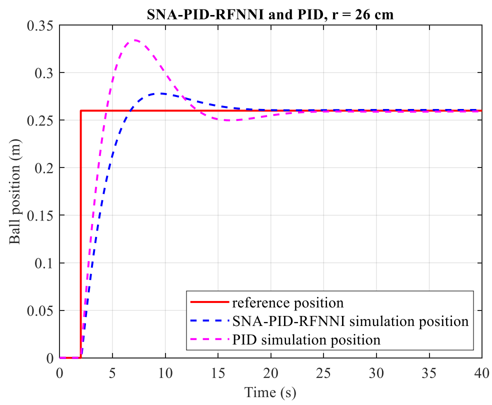


Figure 10. Position response with r =26 cm

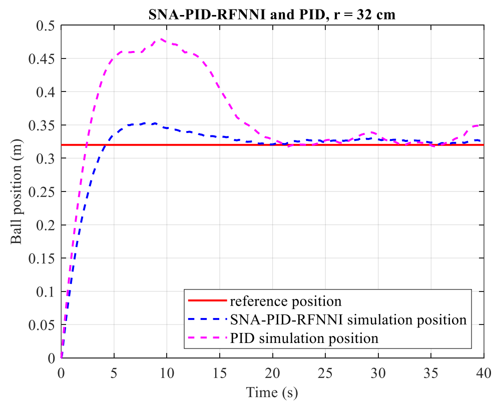


Figure 11. Position response with r =32 cm

Under the condition of no noise (Figure 12), both SNA-PID-RFNNI and PID controllers achieve zero convergence errors. However, the SNA-PID-RFNNI controller gives smoother and faster stability, with smaller oscillation amplitude after the transient period. Meanwhile, under the condition of noise (Figure 13), the PID controller clearly demonstrates the influence of noise, exhibiting strong oscillation errors and slow stabilization. In contrast, the SNA-PID-RFNNI controller still maintains good stability, minor errors, and less oscillation. The Jacob signal of the SNA-PID-RFNNI controller under noise-free and noise conditions is shown in Figures 14 and 15 [23]–[25].

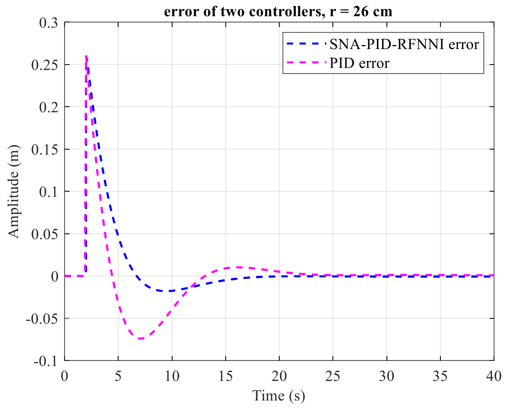


Figure12. Error of two controllers with r =26 cm

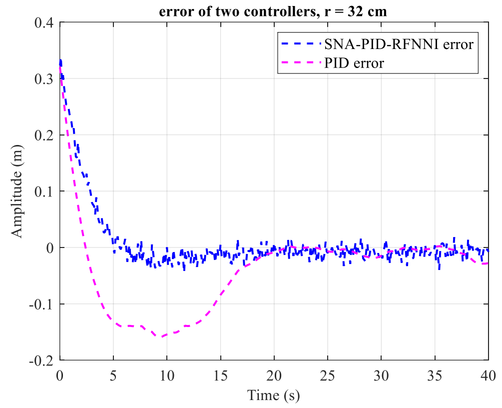


Figure13. Error of two controllers with r =32 cm

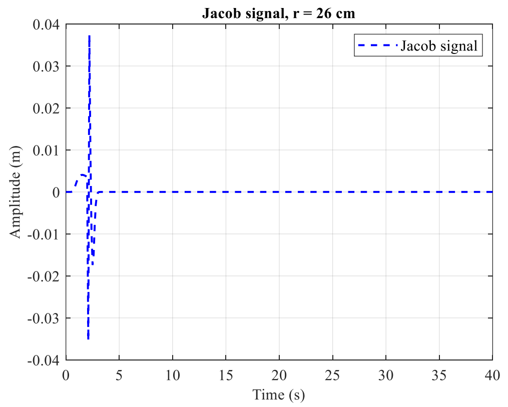


Figure 14. Jacob signal with r =26 cm

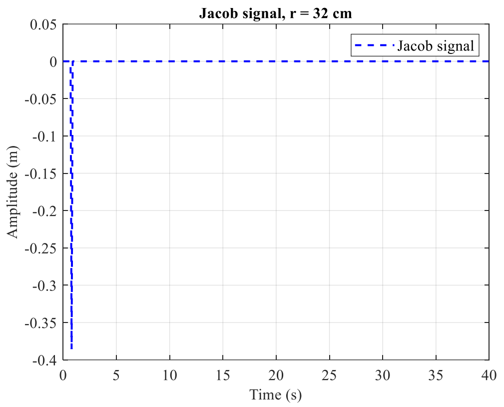


Figure 15. Jacob signal r =32 cm

The results in Table 2 show a clear difference in performance between the two controllers. This improvement reflects the intelligent control system's adaptive and online learning capabilities. Specifically, using single neurons allows for flexible adjustment of the PID coefficients Kp, Ki, Kd in real time, allowing the system to respond faster to state changes. In addition, the RFNNI plays a role in learning and accurately modeling the nonlinear behavior of the rod and ball system, thereby providing feedback information to optimize the controller output. Lower overshoot demonstrates effective control through adjustment, reducing oscillation, thereby helping the system operate more stably. At the same time, shortening the settling time shows that the system can reach equilibrium in a shorter time, which is essential for applications that require fast response and high accuracy.

Table 2. Compare some quality indicators between the two controllers

Controllers	Rise time (s)	Peak time (s)	Settling error (m)	Overshoot (%)	Settling time (s)
PID	2	9	0.018	44.36	22
SNA-PID-RFNNI	4	8	0.0021	5.85	17

5. CONCLUSION

This study presented a control method for a beam and ball system using the SNA-PID-RFNNI controller. The proposed method not only overcomes the limitations of classical PID control in nonlinear systems but also improves adaptability and accuracy in the control process. Simulation results on the MATLAB/Simulink platform confirm that the SNA-PID-RFNNI controller is capable of significantly improving the control quality compared to the traditional PID controller: shorter settling time and peak time, lower overshoot and settling error. These improvements demonstrate the effectiveness of the intelligent control model in handling nonlinear and noisy dynamic systems. In the future, the next step will continue to experimentally deploy the system on real hardware with the MEGA2560 board and IR sensor, to assess the effectiveness and practical implementability of the proposed approach.

FUNDING INFORMATION

The authors declare that no funding was received for this work.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Minh-Thanh Le	✓	✓	✓		✓	✓	✓		✓		✓		✓	
Chi-Ngon Nguyen	✓	✓		✓		✓		✓		✓		✓		✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

DATA AVAILABILITY

The authors confirm that the data supporting the findings of this study are available within the article and its supplementary materials.

REFERENCES

[1] T. Trapp and W. Selby, "Ball and beam balance final report," in *Amazon Web Services*, 2009, pp. 1–28.

[2] V.-H.-L. Tran *et al.*, "Backstepping control for ball and beam: simulation and experiment," *Journal of Fuzzy Systems and Control*, vol. 3, no. 1, pp. 30–38, 2025, doi: 10.59247/jfsc.v3i1.275.

[3] C. N. Nguyễn, V. T. Nguyễn, and T. H. P. Trần, "Điều khiển giám sát hệ cầu cân bằng với thanh và bóng dùng mạng nơ-ron hàm cơ sở xuyên tâm," *Can Tho University Journal of Science*, vol. 58, no. 3, pp. 26–35, 2022, doi: 10.22144/ctu.jvn.2022.083.

[4] B. Meenakshipriya and K. Kalpana, "Modelling and control of ball and beam system using coefficient diagram method (CDM) based PID controller," *IFAC Proceedings Volumes*, vol. 47, no. 1, pp. 620–626, 2014, doi: 10.3182/20140313-3-IN-3024.00079.

[5] A. Kharola and P. P. Patil, "Neural fuzzy control of ball and beam system," *International Journal of Energy Optimization and Engineering*, vol. 6, no. 2, pp. 64–78, 2017, doi: 10.4018/IJEOE.2017040104.

[6] N. S. A. Aziz, R. Adnan, and M. Tajjudin, "Design and evaluation of fuzzy PID controller for ball and beam system," in *2017 IEEE 8th Control and System Graduate Research Colloquium*, 2017, pp. 28–32, doi: 10.1109/ICSGRC.2017.8070562.

[7] X. B. -Chang, W. J. -Zhang, and C. Y. -Kun, "An improved single neuron adaptive PID control algorithm," in *2009 Fifth International Conference on Natural Computation*, 2009, pp. 558–562, doi: 10.1109/ICNC.2009.604.

[8] C.-H. Lee and C.-C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 4, pp. 349–366, 2000, doi: 10.1109/91.868943.

[9] M. T. Nguyen *et al.*, "Method of sliding mode control for ball – beam systems," *Journal of Technical Education Science*, vol. 11, no. 4, pp. 37–42, 2016.

[10] N. N. A. Quan and H. D. Binh, "Design and experimental evaluation of a PID controller for stabilization of the ball and beam balancing system," *Duy Tan University Journal of Science & Technology*, vol. 7, no. 38, pp. 65–73, 2020.

[11] M. Keshmiri, A. F. Jahromi, A. Mohebbi, M. H. Amoozgar, and W.-F. Xie, "Modeling and control of ball and beam system using model based and non-model based control approaches," *International Journal on Smart Sensing and Intelligent Systems*, vol. 5, no. 1, pp. 14–35, 2012, doi: 10.21307/ijssis-2017-468.

[12] M. Tajjudin, S. A. Aziz, N. Ishak, M. H. F. Rahiman, and R. Adnan, "Fuzzy PID tracking performance for ball and beam system," in *2017 IEEE Conference on Systems, Process and Control*, 2017, pp. 100–104, doi: 10.1109/SPC.2017.8313029.

[13] S. Urut, A. Gokcen, M. U. Soydemir, and S. Sahin, "Design and control of ball and beam system using PID control," in *EGE 11th International Conference on Applied Sciences*, 2024, pp. 651–659.
[14] R. Tipsuwanpom, T. Runghimmawan, S. Intajag, and V. Krongratana, "Fuzzy logic PID controller based on FPGA for process control," in *2004 IEEE International Symposium on Industrial Electronics*, 2004, pp. 1495–1500, doi: 10.1109/ISIE.2004.1572035.
[15] W. Wang and Z. Bai, "Performance analysis of an improved single neuron adaptive PID control," in *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, 2010, pp. 22–25, doi: 10.1109/IITSI.2010.18.
[16] X. Tang, T. Huang, X. Liu, and J. Wang, "Application of PID with single-neuron adaptive control in liquid level control," in *2009 Third International Conference on Genetic and Evolutionary Computing*, 2009, pp. 533–536, doi: 10.1109/WGEC.2009.66.
[17] C.-M. Lin and C.-F. Hsu, "Identification of dynamic systems using recurrent fuzzy neural network," in *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, 2001, pp. 2671–2675, doi: 10.1109/NAFIPS.2001.943645.
[18] M.-G. Zhang and W.-H. Li, "Single neuron PID model reference adaptive control based on RBF neural network," in *2006 International Conference on Machine Learning and Cybernetics*, 2006, pp. 3021–3025, doi: 10.1109/ICMLC.2006.258358.
[19] L. M. Thanh, L. H. Thuong, P. T. Loc, and C.-N. Nguyen, "Delta robot control using single neuron PID algorithms based on recurrent fuzzy neural network identifiers," *International Journal of Mechanical Engineering and Robotics Research*, vol. 9, no. 10, pp. 1411–1418, 2020, doi: 10.18178/ijmerr.9.10.1411-1418.
[20] P. Bhounsule, G. Chiou, A. Plascencia, and T. Rowe, "Balancing a ball and beam with PID," in *Proceeding of Department of Mechanical Engineering, The University of Texas*, 2016, pp. 1–15.
[21] P. Jitkhamheang, N. Wongvanich, and W. Tangsrirat, "Design of RBF-based adaptive gain fuzzy sliding mode control for uncertain ball and beam system," in *2024 9th International Conference on Business and Industrial Research (ICBIR)*, 2024, pp. 286–291, doi: 10.1109/ICBIR61386.2024.10875717.
[22] L. M. Thanh, L. H. Thuong, P. T. Tung, C.-T. Pham, and C.-N. Nguyen, "Evaluating the quality of intelligent controllers for 3-DOF delta robot control," *International Journal of Mechanical Engineering and Robotics Research*, vol. 10, no. 10, pp. 542–552, 2021, doi: 10.18178/ijmerr.10.10.542-552.
[23] G. Liqing and L. Yongxin, "Design of BP neural network controller for ball-beam system," in *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference*, 2016, pp. 1087–1091, doi: 10.1109/IMCEC.2016.7867379.
[24] W. Wei and P. Xue, "A research on control methods of ball and beam system based on adaptive neural network," in *2010 International Conference on Computational and Information Sciences*, 2010, pp. 1072–1075, doi: 10.1109/ICCIS.2010.265.
[25] M. L. Thanh, L. H. Thuong, P. T. Tung, and C.-N. Nguyen, "Improvement of PID controllers by recurrent fuzzy neural networks for delta robot," in *Intelligent Communication, Control and Devices*, 2021, pp. 263–275, doi: 10.1007/978-981-16-1510-8_27.

BIOGRAPHIES OF AUTHORS

Minh-Thanh Le received his Bachelor of Electrical and Electronic Engineering Technology from Cuu Long University in 2004, his master's degree in Automation in 2011, and his Ph.D. degree in Control and Automation Engineering in 2024 from Ho Chi Minh City University of Transport. He is currently a lecturer in the Faculty of Electrical and Electronic Engineering, Vinh Long University of Technology and Education. His main research interests are power electronics, motor control, application-oriented robot control, neural networks, and artificial intelligence. He can be contacted at email: thanhlm@vlute.edu.vn.

Chi-Ngon Nguyen received his bachelor and master degrees in Electronic Engineering from Can Tho University and the Vietnam National University, Ho Chi Minh City University of Technology, in 1996 and 2001, respectively. The degree of Ph.D. was awarded by the University of Rostock, Germany, in 2007. Since 1996, he has worked at the Can Tho University. Currently, he is a professor in automation at Department of Automation Technology. He is also a national-distinguished lecturer and a secretary of the national council for professorships in electrical-electronics and automation. His research interests are AI applications, intelligent control, medical control, pattern recognition, classifications, computer vision, and agricultural automation. He can be contacted at email: ncngon@ctu.edu.vn.