

Analysis of Linear and Intelligent Control for Balancing Pendubot System

Minh-Duy Tran¹, Diep-Thuy-Duong Le^{2,*}, Hong-Phuoc Phan³, Hoang-Viet Vo⁴, Dang-Quang-Tinh Ngo⁵,
Ngoc-Duy Nguyen⁶, Tan-Phat Nguyen⁷, Nhat-Linh Tran⁸, Thanh-An Vo⁹, Thi-Thanh-Hoang Le¹⁰
^{1, 6, 9} Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology and Education
(HCMUTE), Vietnam

² Viet An Environment Technology JSC, Ho Chi Minh City, Vietnam

^{3, 4, 5, 7, 8} Faculty of International Training, Ho Chi Minh City University of Technology and Education (HCMUTE), Vietnam

Email: ¹ 21151087@student.hcmute.edu.vn; ² lediepthuyduong@gmail.com; ³ 21151430@student.hcmute.edu.vn;

⁴ 21151436@student.hcmute.edu.vn; ⁵ 21151056@student.hcmute.edu.vn; ⁶ 20151345@student.hcmute.edu.vn;

⁷ 21151041@student.hcmute.edu.vn; ⁸ 21151031@student.hcmute.edu.vn; ⁹ 20142275@student.hcmute.edu.vn;

¹⁰ hoanglth@hcmute.edu.vn

*Corresponding Author

Abstract—Pendubot is a typical under-actuated SIMO control system, commonly used in research on control algorithms. Rather than focusing on analyzing a single control algorithm, this paper provides an overview of control efficiency as well as differences between algorithms through analytical assessments. In this study, the authors analyzed algorithms including feedback linearization (a linear algorithm), LQR – optimal control (a linear algorithm), and fuzzy control (an intelligent algorithm) to stabilize the model at the equilibrium position of the TOP position – where both bars of the system stand upright in the opposite direction to gravity. The genetic algorithm (GA) is used to optimize control parameters for the model. These algorithms are simulated in MATLAB/Simulink, and the simulation results are compared, concluding that the LQR control algorithm is the most optimal for balancing this model.

Keywords—Fuzzy Control; LQR Control; Intelligent Control; Linear Control Pendubot; SIMO System

I. INTRODUCTION

Pendubot, fully named Pendulum Robot, is a classic system with significant importance in research and education within control automation laboratories. With Pendubot, researchers can explore fundamental concepts such as nonlinear dynamics, dynamic system theory, and control algorithms. Pendubot differs from previous mechatronic systems, such as Acrobot [1] and Furuta's inverted pendulum [2]. Once the theories applied to standard models (such as the Pendubot or rotating inverted pendulum) are fully developed and scientists can apply similar algorithms to real-world systems. The study of this system can be further expanded by adding additional links to create higher-order systems [3]. However, this approach is not widely applicable in practice, which is why 2-link Pendubot remains the most frequently studied subject. In reference [4], the authors compared sliding and fuzzy sliding algorithms for Pendubot. However, these algorithms have complex structures and are difficult to widely implement for algorithm training purposes. In reference [5], fuzzy-ANFIS algorithm was applied to compare the LQR algorithm with the fuzzy control developed later, showing that the fuzzy-ANFIS algorithm

developed from the original controller is equivalent. Nevertheless, a comparison between a non-ANFIS fuzzy controller and a separate LQR controller is necessary. A controller based on the LQR algorithm is introduced in reference [6]. Interestingly, in this study, the controller is embedded in a PLC, which is widely used in industrial applications but not common in small-scale balancing robot control. However, no further comparisons with other algorithms for LQR controllers are provided in this research. The potential of intelligent controllers is substantial [7]. Additionally, the sliding mode control algorithm has also been applied to Pendubot, as mentioned in the reference [8]. However, regarding control quality, further comparisons between linear and intelligent algorithms are needed to provide a comprehensive assessment. Once balancing algorithms have stabilized the system, performing Swing-Up control becomes easier. Several studies have shown that combining algorithms for Swing-Up control is entirely feasible, including sliding mode control [9], energy-based methods [10], and even Fuzzy logic [11].

As mentioned above, there are many studies on balancing Pendubot, but no paper has yet compared and analyzed different balancing control methods with each other. Most focus on a single control algorithm. Recognizing this, in this paper, the authors selected one representative controller from each of the three most common control methods (linear, nonlinear, and intelligent) for a general comparison. From simulation results for balancing Pendubot in the TOP position using Feedback Linearization (linear control algorithm), Fuzzy (intelligent control algorithm), and LQR (linear control algorithm) methods, the research team concludes that the LQR algorithm is the optimal balancing control for Pendubot.

This paper is organized as follows: Section 2 describes the dynamic mathematical equations and control algorithms of the Pendubot system. The simulation results are presented in Section 3, and conclusions are provided in Section 4.

II. MATHEMATICAL EQUATIONS AND SIMULATION

A. Mathematical Equations

The dynamic equations of the system based on the Euler-Lagrange formulation are as follows [12]:

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) = \tau \quad (1)$$

where:

$$M(q) = \begin{bmatrix} \beta_1 + \beta_2 + 2\beta_6 \cos q_2 & \beta_2 + \beta_3 \cos q_2 \\ \beta_2 + \beta_3 \cos q_2 & \beta_2 \end{bmatrix} \quad (2)$$

$$V(q, \dot{q}) = \begin{bmatrix} -\beta_3 \sin q_2 \dot{q}_2 & -\beta_3 \sin q_2 \dot{q}_2 - \beta_3 \sin q_2 \dot{q}_1 \\ \beta_3 \sin q_2 \dot{q}_1 & 0 \end{bmatrix} \quad (3)$$

$$G(q) = \begin{bmatrix} \beta_4 g \cos q_1 + \beta_5 \cos(q_1 + q_2) \\ \beta_5 \cos(q_1 + q_2) \end{bmatrix} \quad (4)$$

$$\begin{aligned} \beta_1 &= m_1 l_1^2 + m_2 l_1^2 & \beta_2 &= m_2 l_{c2}^2 & \beta_3 &= m_2 l_1 l_2 \\ \beta_4 &= m_1 l_1 + m_2 l_1 & \beta_5 &= m_2 l_2 g & \beta_6 &= m_1 m_2 \end{aligned} \quad (5)$$

The external torque is converted to the input voltage value as follows:

$$\tau_m = a_1 u - a_2 \dot{q}_1 - a_3 \ddot{q}_1 \quad (6)$$

With:

$$a_1 = \frac{K_t}{R_m} \quad a_2 = \frac{K_t K_b}{R_m} \quad a_3 = J_m \quad (7)$$

Solving equation **Error! Reference source not found.**, we obtain:

$$\ddot{q}_1 = \frac{-1}{2\beta_3^2 \cos^2 q_2 - 2\beta_1 \beta_2 + 4\beta_2 \beta_3 \cos q_2 - 4\beta_2 \beta_6 \cos q_2} [2\beta_2 \beta_3 \sin q_2 (\dot{q}_1^2 + \dot{q}_2^2 + 2\dot{q}_1 \dot{q}_2) + \beta_3^2 \sin(2q_2) \dot{q}_1^2 - 2\beta_2 \beta_4 g \cos q_1 + \beta_3 \beta_5 (\cos q_1 + \cos(q_1 + 2q_2)) - 2\beta_2 \tau_1] \quad (8)$$

$$= \frac{1}{\beta_3^2 \cos^2 q_2 - \beta_1 \beta_2 + 2\beta_2 \beta_3 \cos q_2 - 2\beta_2 \beta_6 \cos q_2} (\beta_2 + \beta_3 \cos q_2)(\tau_1 - \beta_5 \cos(q_1 + q_2)) - \beta_4 g \cos q_1 + \beta_3 \sin q_2 (\dot{q}_2^2 + 2\dot{q}_1 \dot{q}_2) + (\beta_3 \sin q_2 \dot{q}_1^2 + \beta_5 \cos(q_1 + q_2)) (\beta_1 + \beta_2 + 2\beta_6 \cos q_2) \quad (9)$$

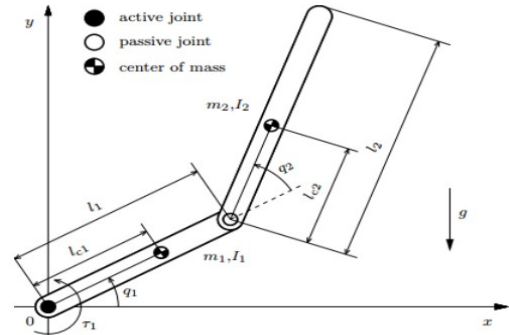


Fig. 1. Structure of the pendubot description [13]

In this paper, the authors focus on examining the Pendubot system at the TOP position. x_1 and x_3 correspond to the angular deviations of link 1 and link 2 relative to the Oy axis, respectively. x_2 and x_4 are the angular velocities of link 1 and link 2, respectively. The state variables are redefined as follows:

$$\begin{aligned} x_1 &= q_1 - \frac{\pi}{2}; & x_2 &= \dot{x}_1 = -\dot{q}_1 \\ x_3 &= q_2; & x_4 &= \dot{x}_3 = \dot{q}_2 \end{aligned} \quad (10)$$

B. Control Algorithm

1) Feedback Linearization

Feedback linearization control is an approach used to design controllers for nonlinear systems, where the control signal appears linearly in the state equations. This technique transforms a nonlinear system into a linear one by using state feedback and coordinate transformation. This section refers to [7], pages 307–312. Consider the nonlinear SISO system described by the following state equation:

$$\begin{cases} \dot{x} = f(x) + g(x)u \\ y = h(x) \end{cases} \quad (11)$$

According to the control rule combined with Lie derivatives, the system output is determined by the following formula:

$$\frac{d^n y}{dt^n} = L_f^n h(x) + L_g L_f^{n-1} h(x) u \quad (L_g L_f^{n-1} h(x) \neq 0) \quad (12)$$

where:

$$\begin{aligned} \alpha(x) &= L_f^n h(x); & \alpha(x) &= L_f^n h(x); \\ \beta(x) &= L_g L_f^{n-1} h(x) \end{aligned} \quad (13)$$

System linearization control signal:

$$u(x) = \frac{v(t) - \alpha(x)}{\beta(x)} \quad (14)$$

Where $v(t)$ is the signal vector of the system input

$$v = -Kz \quad (15)$$

where:

$K = [K_1 \ K_2 \ K_3 \ K_4]$ is a matrix with positive

coefficients $z = \begin{bmatrix} h(x) \\ L_f^1 h(x) \\ L_f^2 h(x) \\ L_f^3 h(x) \end{bmatrix}$ is a state variable.

2) Fuzzy Control

Fuzzy logic theory was first introduced in [14]. This theory addresses problems that closely resemble human reasoning. Unlike traditional logic, fuzzy logic produces results (outputs) as one of the countless real values between 0 and 1. Moreover, fuzzy logic provides an intermediate value, depending on the degree of the input conditions. The application of fuzzy algorithms to underactuated systems was presented in reference [15]. The fuzzy rules used in this paper are based on 81 fuzzy rules mentioned in reference [7] page 182, with adjustments made to the reference frame of Pendubot to match the author's chosen reference frame for inverted pendulum on cart (IPC) system. In this study, the authors only adjusted the system from an IPC to a Pendubot without modifying the fuzzy logic rules.

The operating principle of a fuzzy controller can be described as follows: Fuzzification is the process where input values, such as error and error rate in control, are fed into the fuzzy controller and converted into fuzzy values. The controller then applies fuzzy rules from its rule base to evaluate the situation and determine the appropriate control actions. Through fuzzy inference, the controller calculates the control outcome based on the applied rules, with fuzzy values computed to make a decision. Finally, defuzzification converts this fuzzy control result back into a specific numerical value that the system can use. The operating principle of fuzzy control is illustrated in Fig. 2.

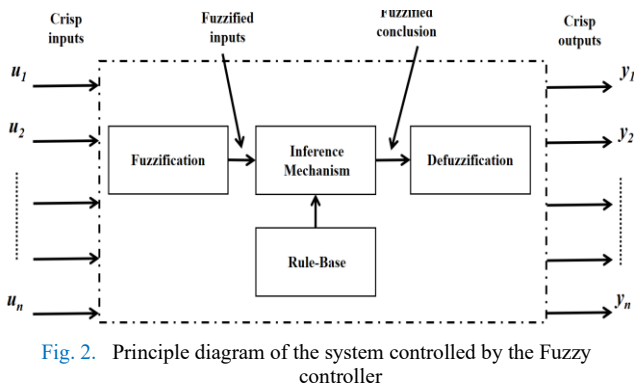


Fig. 2. Principle diagram of the system controlled by the Fuzzy controller

3) LQR

Consider a system with a control signal ($u \neq 0$):

$$\dot{x} = Ax + Bu \tag{16}$$

Refer to [6] the system control signal follows (17), where u is the input of the Controller, x is the matrix of state variable values, K is the feedback matrix, and y is the output of the Controller, which represents the control signal of the system—either voltage or torque. The block

diagram of the system controlled by the LQR Controller is illustrated in Fig. 3.

$$u(t) = -Kx(t) \tag{17}$$

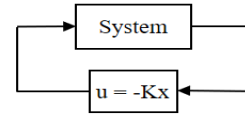


Fig. 3. Block diagram of the principle of the system controlled by the LQR controller

Weighting matrix Q:

$$Q = \begin{bmatrix} Q_1 & 0 & \dots & 0 \\ 0 & Q_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q_n \end{bmatrix} \tag{18}$$

Energy coefficient R ($R > 0$)

Solving the Riccati equation, the solution is matrix S:

$$A^T S + SA - SBR^{-1}B^T S + Q = 0 \tag{19}$$

Control matrix K:

$$K = R^{-1}B^T S \tag{20}$$

Calculate the matrix K in MATLAB:

$$K = dlqr(Ad, Bd, Q, R) \tag{21}$$

C. Simulation Program

The selection of parameters for each controller significantly impacts the control quality of the system. In practice, there are various methods for finding optimal parameters for controllers with typical examples being PSO (Particle Swarm Optimization) as mentioned in the reference [16] and GA presented in reference [17]. In this paper, the authors utilize GA to find the optimal parameters for all three controllers. For the LQR Controller, the control matrix K_{LQR} needs to be optimized. K_{LQR} is influenced by the weighting matrix Q and the energy coefficient R, so GA is used here to determine the best coefficients Q and R. For the Feedback Linearization Controller, the coefficients that directly affect the system response are contained in the matrix K_{FLC} . For the Fuzzy Controller, it is essential to accurately determine the preprocessing and postprocessing coefficients to ensure that the system output approaches zero as quickly and stably as possible. In this study, GA applied to Fuzzy is only used to optimize the preprocessing coefficients (K_1, K_2, K_3, K_4) and postprocessing coefficients (K_5), without intervening in or altering the fuzzy logic rules.

Table 1. Initial state variables

Parameter	x_{1init}	x_{2init}	x_{3init}	x_{4init}
Value	0.05	0.05	0.05	0.05

Table 2. System parameters

Symbol	Value	Unit	Description
m_1	0.16	kg	Mass of link 1
l_1	0.2	m	Length of link 1
l_{c1}	0.1	m	Distance from active joint to center of mass of link 1
I_1	0.00222	kgm ²	Moment of inertia of link 1
m_2	0.066	kg	Mass of link 2
l_2	0.22	m	Length of link 2
l_{c2}	0.11	m	Distance from passive joint to center of mass of link 2
I_2	0.00106	kg.m ²	Moment of inertia of link 2
K_t	0.0198	Nm	Torque constant
K_b	0.0198	V	Back EFM constant (Force constant of motor)
R_m	6.835	Ω	Resistor of rotor
J_m	0.000134	kg.m ²	Moment of inertia of rotor
C_m	0.000048	Nm	Viscous friction constant

Fig. 4, Fig. 5, and Fig. 6 represent the simulation of the control algorithms using Matlab/Simulink software. In the Pendubot block, the equations are those outlined in section A of II. Similarly, the blocks for Feedback Linear Controller, Fuzzy Controller, and LQR Controller contain the equations presented in section B of II. The output response of the system for each algorithm is observed through the Scope block. All results are presented in section III.

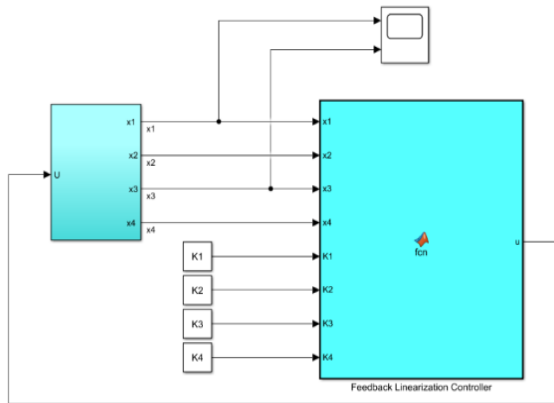


Fig. 4. Simulation of the feedback linear controller (FLC) for the Pendubot system

Parameters of the matrix K corresponding to the K values in the simulation program:

$$K_{FLC} = [62.111 \ 941.4010 \ 96.5010 \ 62.5310]$$

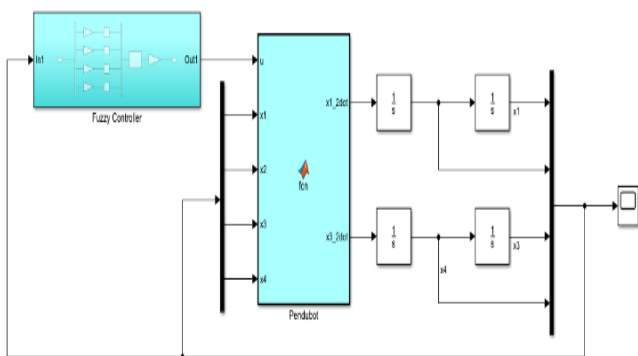


Fig. 5. Fuzzy controller program for the Pendubot

Pre-processing and post-processing parameters:

$$K_1 = 4.6; K_2 = 1.4; K_3 = -4.3; K_4 = -2; K_5 = -20$$

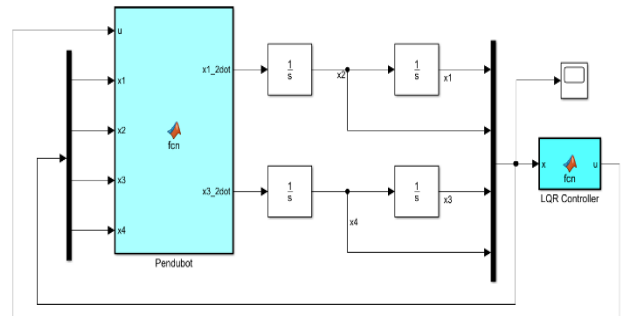


Fig. 6. LQR controller program for the Pendubot system

Control matrix for LQR control:

$$K_{LQR} = 10^3 \times [-0.0313 \ 0.1970 \ 3.5168 \ 0.5409]$$

III. RESULT

A. Simulation results of the linear feedback controller

The simulation results of the linear feedback controller are shown in Fig. 7 and Fig. 8. Link 1 stands upright and stops moving after 0.6s. Link 2 stands upright and stops moving after 0.6s. Thus, the linear feedback controller stabilizes the Pendubot system after 0.6s. The system as shown in the figure has oscillations ranging from 0.003 (rad) to -0.002 (rad) with a steady-state error of 0.8 degrees. The settling time is 0.6s. The results indicate that the two links can stabilize well but the slow settling time is a limitation in balancing the system. From this simulation, it can be observed that when experimenting with other external influencing factors, the feedback linearization algorithm commonly used for SISO systems may not be optimal when applied to a SIMO system like the Pendubot.

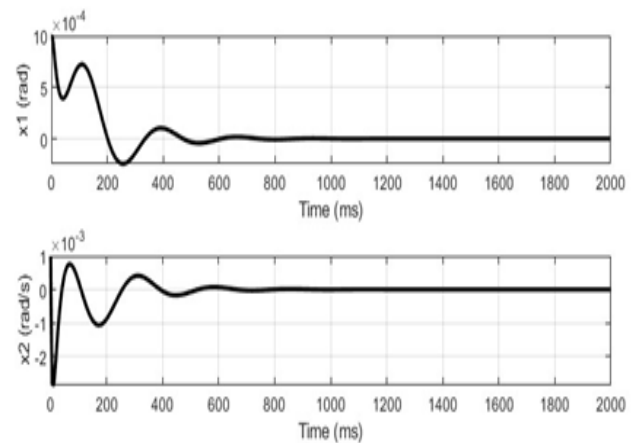


Fig. 7. Simulation results of the linear feedback controller for the angle and angular velocity of link 1

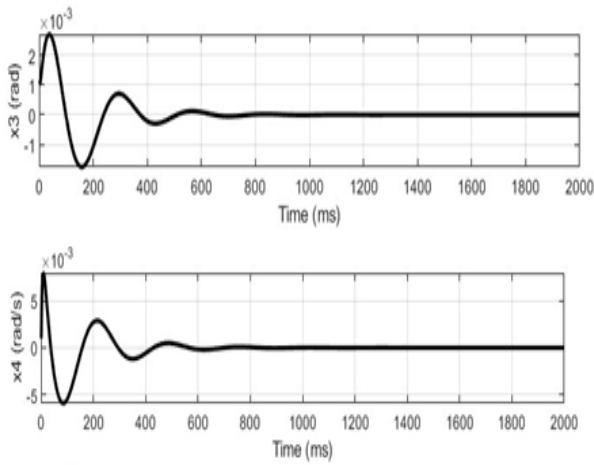


Fig. 8. Simulation results of the linear feedback controller for the angle and angular velocity of link 2

B. Simulation results of the Fuzzy controller

The simulation results of the Fuzzy controller are shown in Fig. 9 and Fig. 10. Link 1 stands upright and stops moving after 1.8s. Link 2 stands upright and stops moving after 1.4s. Thus, the Fuzzy controller balances Pendubot after 1.8s. The system as shown in the figure has a long settling time. During the process of two links reaching the balanced position, the system exhibits strong oscillations, with the amplitude ranging from 0.007 (rad) to -0.007 (rad), and a steady-state error of 1.3 degrees. The results indicate that while two links can stabilize, large steady-state error makes the system susceptible to imbalance when external forces are applied. Large oscillation amplitude during the balancing process significantly slows down settling time, and such continuous oscillations would make experimental implementation challenging. Based on these simulation results, the intelligent control algorithm is not yet optimal for balancing the control of Pendubot.

C. Simulation results of the LQR controller

The simulation results of the LQR controller are shown in Fig. 11 and Fig. 12. Link 1 stands upright and stops moving after 0.4s. Link 2 stands upright and stops moving after 0.15s. Thus, the LQR controller balances the Pendubot system after 0.4s. For the LQR controller, the simulation results show that the system maintains better stability compared to the two control algorithms mentioned earlier. The two links exhibit minimal oscillation during control, leading to a fast settling time of 0.4s and a low steady-state error of 0.2 degrees. Although there is a significant overshoot at the beginning of the control process, the system stabilizes quickly, so it does not significantly affect control quality. The results demonstrate that LQR is a suitable control algorithm for experimental applications with the Pendubot. Given its high control quality, LQR can be combined with other control algorithms to create Swing-Up controllers such as SMC-LQR, EBM-LQR, and FLC-LQR.

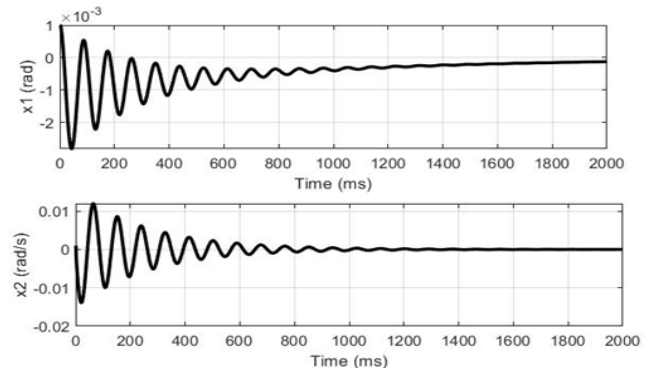


Fig. 9. Simulation results of the Fuzzy controller for the angle and angular velocity of link 1

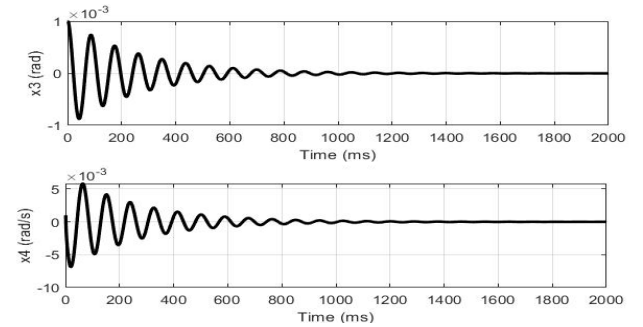


Fig. 10. Simulation results of the Fuzzy controller for the angle and velocity of bar 2

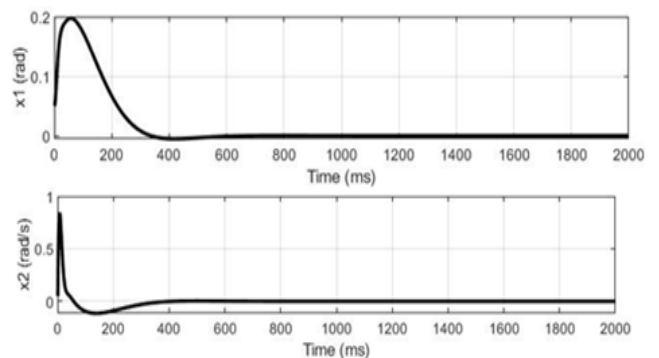


Fig. 11. Simulation results of the LQR controller for the angle and velocity of link 1

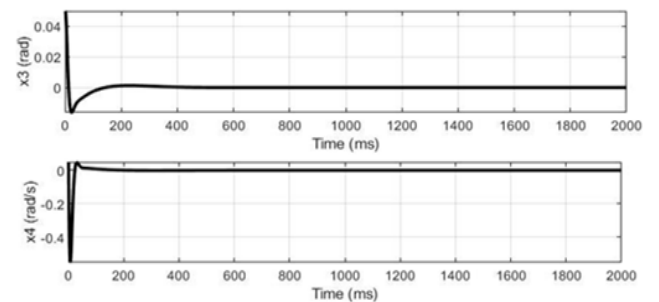


Fig. 12. Simulation results of the LQR controller for the angle and velocity of link 2

Table 3. Comparison of Results

Algorithm	Balancing Time (s)
FLC	0.6
Fuzzy	1.8
LQR	0.4

IV. CONCLUSION

Through simulation and comparison, it was observed that all controllers balanced Pendubot. Among them, the LQR controller stabilized the system fastest and with the least oscillation. For the Fuzzy controller, the parameters found provided the best response for the best output, but there were still too many oscillations, and the response time was slower compared to the other two controllers (1.8s). For these reasons, it is clear that using the LQR algorithm offers the best control quality and is the most feasible option when tested on an experimental model.

ACKNOWLEDGEMENT

This contribution belongs to the project, which is funded by Ho Chi Minh City University of Technology and Education (HCMUTE), for students of HCMUTE in the year 2025. We also want to give thanks to the PhD. Van-Dong-Hai Nguyen (HCMUTE) helped us in theory.

REFERENCES

- [1] M.W.Spong, "Swing up control of the Acrobot" *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 1994, <https://doi.org/10.1109/ROBOT.1994.350934>.
- [2] K. Yoshida, "Swing-up control of an inverted pendulum by energy-based methods" *Proceedings of the 1999 American Control Conference*, 1999, <https://doi.org/10.1109/ACC.1999.786297>.
- [3] N.P.Hieu *et al.*, "Optimal control for 3-linked Pendubot," *Tien Giang University Science Journal*, vol. 9, 2020, <https://doi.org/10.1010/tckh.v1i9.12>.
- [4] M.-D. Tran *et al.*, "A Comparative Study between Sliding Mode and Fuzzy Sliding Mode Control for Pendubot," *Serbian Journal of Electrical Engineering*, vol. 20, pp. 361-390, 2023, <https://doi.org/10.2298/SJEE2303361T>.
- [5] N. T. Nguyen *et al.*, "The Pendubot control uses a linear omnidirectional form based on fuzzy logic," *Dong Thap University Science Journal*, vol. 38, pp. 89-93, 2019, <https://doi.org/10.52714/dthu.38.6.2019.704>.
- [6] V.-D. Nguyen *et al.*, "Optimal Control of Pendubot System Using SIMATIC S7-1200," *International Journal of Robotica & Management*, vol. 28, no. 1, pp. 43-52, 2023, <https://doi.org/10.24193/rm.2023.1.6>.
- [7] S. N. Vassilyev, A. Y. Kelina, Y. I. Kudinov, and F. F. Pashchenko, "Intelligent control systems," *Procedia Computer Science*, vol. 103, pp. 623-628, 2017, <https://doi.org/10.1016/j.procs.2017.01.088>.
- [8] Dong Sang Yoo, "Balancing control for the pendubot using sliding mode," *IEEE ISR 2013*, pp. 1-4, 2013, <https://doi.org/10.1109/ISR.2013.6695735>.
- [9] C. V. Kien *et al.*, "Swing Up and Balancing Implementation for the Pendubot Using Advanced Sliding Mode Control," *International Conference on Artificial Intelligence and Control Automation*, 2015, <https://doi.org/10.2991/eame-15.2015.108>.
- [10] I. Fantoni *et al.*, "Energy Based Control of the Pendubot," *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, vol. 45, 2000, <https://doi.org/10.1109/9.847110>.
- [11] X. Q. Ma *et al.*, "A new fuzzy approach for swing up control of Pendubot," *Proceedings of American Control Conference*, 2002, <https://doi.org/10.1109/ACC.2002.1023149>.
- [12] D. J. Block *et al.*, "Mechanical Design and Control of the Pendubot," *SAE Transactions*, vol. 104, pp. 36-43, 1995, <https://doi.org/10.2991/eame-15.2015.108>.
- [13] Martin Gulan *et al.*, "Tracking Control of Unforced and Forced Equilibrium Positions of the Pendubot System: A Nonlinear MHE and MPC Approach," *Actuators*, vol. 12, no. 9, 2023, <https://doi.org/10.3390/act12090343>.
- [14] L.A.Zadeh *et al.*, "Fuzzy Sets," *Information and control*, vol. 8, pp. 338-353, 1965, [https://doi.org/10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X).
- [15] M. Shibli *et al.*, "Direct Adaptive Control for Underactuated Mechatronic Systems using Fuzzy Systems and Neural Networks: A Pendubot Case," *Canadian Conference on Electrical and Computer Engineering*, 2007, <https://doi.org/10.1109/CCECE.2006.277489>.
- [16] S. Yuan *et al.*, "Research on Control Problem of PenduBot Based on PSO Algorithm," *International Conference on Computational Intelligence and Natural Computing*, 2009, <https://doi.org/10.1109/CINC.2009.132>.
- [17] Trong-Bang Tran *et al.*, "Trajectory Tracking using LQR Control for Pendubot: Simulation and Experiment," *JFSC*, vol.2, no.1, 2007, <https://doi.org/10.59247/jfsc.v2i1.163>.