CONTROL ROTARY INVERTED PENDULUM MODEL BALANCING USING AN IMPROVED FUZZY CONTROLLER BASED ON A QUANTUM SWARM OPTIMIZATION ALGORITHM

Thanh-Lam Bui[1,*], Van-Truong Nguyen[1], Ngoc-Tien Tran[1], Duc-Quang Nguyen[1], Tuan-Hung La[1], Minh-Quang Nguyen[1], Trung-Thanh Le[1]

DOI: http://doi.org/ 10.57001/huih5804.2025.265

ABSTRACT

This paper presents an optimization method for fuzzy logic controllers (FLC) utilizing the quantum-behaved particle swarm optimization (QPSO) algorithm. The proposed controller is implemented for balance control of a rotary inverted pendulum, where the parameters of the triangular membership function are fine-tuned to achieve optimal error and transient response time for the system's state variables. By integrating quantum mechanics principles with the particle swarm optimization (PSO) framework, QPSO demonstrates robust capabilities in identifying global and local optima for complex nonlinear and non-differentiable problems. To evaluate the algorithm's optimization performance, simulations were conducted using Matlab software, and the algorithm was implemented on an experimental model. Traditional PSO was also included for comparison. Simulation and experimental results show that QPSO achieves faster convergence and superior search outcomes under identical conditions (both with and without noise), along with improved quality indices compared to PSO.

Keywords: *Quantum-behaved particle swarm optimization, fuzzy logic controller, rotary inverted pendulum.*

[1]School of Mechanical and Automotive Engineering, Hanoi University of Industry, Vietnam

[*]Email: thanhlam710@gmail.com

Received: 10/3/2025

Revised: 05/5/2025

Accepted: 25/7/2025

1. INTRODUCTION

The inverted pendulum system holds significant importance in nonlinear mechanics. It represents numerous industrial challenges and applications, such as external disturbances or nonlinear behaviors under varying conditions [1]. Consequently, it has secured a prominent position as a foundational experimental platform for validating and testing novel ideas in control theory. The inverted pendulum model is highly suitable for simulating state control of space rockets and satellites, automatic aircraft landing systems, aircraft stabilization in turbulent airflow, cabin stabilization on ships, and more. Additionally, this model serves as an initial step toward robot stabilization [2]. The inverted pendulum is one of the most extensively studied problems in control engineering and remains among the most complex dynamic systems [4].

The rotary inverted pendulum is an open-loop unstable and highly nonlinear system, making its control a challenging task. Stabilizing the pendulum rod in an upright position with oscillations in its position is considered a benchmark control problem. This has been addressed by attaching the pendulum to a base that either moves linearly (classical inverted pendulum system) or rotates horizontally on a plane (rotary inverted pendulum system) [3]. Simulating, predicting, or controlling an inverted pendulum system requires an accurate mathematical model of its highly complex dynamics, typically described by differential equations [2]. Furthermore, applying control techniques to such a complex nonlinear system poses challenges in selecting control parameters and ensuring performance efficiency. Achieving optimal performance involves enhancing operational capability, energy efficiency, and stability. For instance, H. El Aiss and R. Orellana [5] modeled the cart-inverted pendulum system and utilized Linear Matrix Inequalities (LMI) to design a fuzzy controller. Their results showed system stabilization within a short period, with responses maintaining amplitude and speed limits. Similarly, Rani et al. [6] employed an ID controller optimized with a Genetic Algorithm (GA), resulting in significantly improved PID performance, faster stabilization, and optimized response. For controlling the rotary inverted pendulum, various methods can be applied, including linear control methods, system linearization, using gain scheduling, fuzzy logic control (FLC), swing-up control, switching control, balance control and trajectory tracking control.

Among various control methods, FLC has emerged as an effective solution due to its ability to emulate human decision-making processes and handle imprecise inputs without requiring an exact mathematical model of the system [7]. FLC utilizes a set of linguistic rules to define the relationship between system states and control actions, making it particularly suitable for controlling nonlinear systems like the inverted pendulum [8]. This approach not only enhances system stability but also provides resilience against parameter variations and external disturbances, which are common in real-world scenarios [9]. To further improve the performance of fuzzy control, optimization algorithms such as neural networks [10], genetic algorithms (GA) [11], and PSO [12] have been applied. Researchers have optimized fuzzy logic controllers using PSO by fine-tuning fuzzy parameters like membership function boundaries and rule weights, combined with the Lyapunov method to ensure global stability for nonlinear systems. Results have shown faster stabilization, reduced oscillations, and improved performance under noisy conditions and environmental parameter variations compared to traditional controllers. A variation of PSO, known as QPSO [13], has also been employed. By updating particles based on quantum behavior [14] rather than traditional position and velocity updates influenced by individual and collective experiences, QPSO exhibits superior global convergence [15] and avoids local optima issues [16]. QPSO has been used to optimize various controllers, such as LQR [17], PID [18], and multi-segment models of nonlinear systems [19], achieving promising results in precise control and system stability under disturbances. For instance, Reddipogu [20] utilized QPSO to enhance global search capability, optimizing the Q and R weights in the LQR cost function. This approach improved system disturbance rejection, ensured rapid and accurate responses under changing load or resistance conditions, and significantly reduced oscillations compared to LQR tuned with conventional PSO.

To further enhance the performance of FLC, the integration with the QPSO algorithm has been introduced. Leveraging the unique strengths of both approaches-the nonlinear signal processing capability of FLC and the parameter optimization power of QPSO-this combination promises to be an effective solution for controlling nonlinear systems. This paper presents a QPSO-enhanced FLC algorithm design, where the control strategy adapts and self-tunes to ensure stability in the rotary inverted pendulum system.

2. ROTARY INVERTED PENDULUM SYSTEM

A typical model of a rotary inverted pendulum can be seen in Fig. 1. The system parameters are represented as follows: the angular position of the arm (θ_1), the angular position of the pendulum rod θ_2), the motor control signal (v_i), the arm length (L_1), the pendulum rod length (L_2), the arm mass (m_1), the pendulum rod mass (m_2), the center of mass of the arm (l_1), the center of mass of the pendulum rod (l_2), the moment of inertia of the arm (J_1), the moment of inertia of the pendulum rod (J_2), the damping coefficient of the arm (b_1), and the damping coefficient of the pendulum rod (b_2).

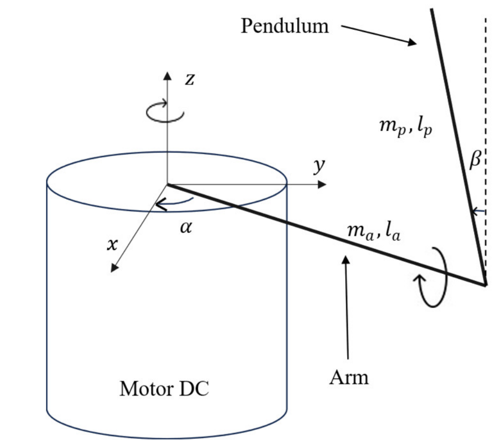


Fig. 1. Rotaty inverted pendulum

According to the Lagrange equation, the dynamics of the Rotary Inverted Pendulum (RIP) system can be expressed as follows:

$$\begin{aligned}
&[\widehat{J_0} + \widehat{J_2}(\theta_2)\ \ m_2 l_1 l_2 \\
&\cos\cos(\theta_2)\ \ m_2 L_2 l_2 \cos(\theta_2\ \widehat{J_2}\ ][\ddot{\theta}_1\ \ddot{\theta}_2\ ] \\
&+\Big[b_1 + \frac{1}{2}\dot{\theta}_2 \widehat{J_2} \sin(2\theta_2)(\frac{1}{2}\dot{\theta}_2 \widehat{J_2}\ \sin\sin(2\theta_2) \\
&\qquad - \ m_2 L_1 l_2 \sin(\theta_2)\ddot{\theta}_2) \\
&\qquad - \frac{1}{2}\dot{\theta}_2\ \widehat{J_2} \sin(2\theta_2) b_2\ \Big][\dot{\theta}_1 \dot{\theta}_2] \\
&+[0\ g m_2 l_2\ \sin\sin(\theta_2)\ \ ] = [\tau\ 0\ ]
\end{aligned} \tag{1}$$

Với $\widehat{J_2} = J_2 + m_2 l_2^2, \widehat{J_0} = J_1 + m_1 l_1^2 + m_2 L_1^2$

In practice, the input to the motor is voltage. Therefore, we need an equation to represent the relationship between the input voltage and the control force calculated from equation (1). This relationship is expressed in the following equation (2):

$$\tau = k_1 v_i + k_2 \dot{\theta}_1 \quad (2)$$

By substituting equation (2) into equation (1), the motion equations of the system can be expressed as follows:

$$\begin{aligned} &\left(A + B\sin^2(\theta_2)\right)\ddot{\theta}_1 + C\cos(\theta_2)\ddot{\theta}_2 + F\dot{\theta}_1 \\ &+B\dot{\theta}_1\dot{\theta}_2 \sin\sin(2\theta_2) - C\dot{\theta}_2 \sin\sin(\theta_2) = Iv_i \end{aligned} \quad (3)$$

$$\begin{aligned} &C\cos(\theta_2)\ddot{\theta}_1 + B\ddot{\theta}_2 - \frac{B}{2}\dot{\theta}_1 \sin\sin(2\theta_2) \\ &+E\dot{\theta}_2 + D\sin(\theta_2) = 0 \end{aligned} \quad (4)$$

Với $A = \widehat{J_0},\ B = \widehat{J_2},\ C = m_2 L_1 l_2,\ D = g m_2 l_2,\ E = b_2,\ F = b_1 - K\_2, I = K_1$

The state-space equations of the system can be derived from equation (3) as follows:

$$\dot{q}_1 = p_1$$

$$\dot{p}_1 = \frac{\begin{matrix} B[-Fp_1 - Bp_1p_2 \sin\sin(2q_2) \\ +Cp_2^2 \sin\sin(q_2)\,] \end{matrix}}{AB + B^2(q_2) - C^2\,(q_2)}$$

$$\dot{q}_2 = p_2$$

$$\begin{aligned} &\dot{p}_2 \\ &= \frac{\begin{matrix} -C\cos(q_2)[-Fp_1 - Bp_1p_2 \sin\sin(q_2) \\ +Cp_2^2 \sin\sin(q_2)\,] \end{matrix}}{AB + B^2(q_2) - C^2\,(q_2)} \\ &\quad \left(A + B\sin^2(q_2)\right)\begin{bmatrix} -\frac{B}{2}p_1^2 \sin\sin(2q_2) \\ +Eq_2 + D\sin(q_2) \end{bmatrix} \\ &- \frac{-CI\cos(q_2)v_i}{AB + B^2(q_2) - C^2\,(q_2)} \end{aligned} \quad (5)$$

3. FUZZY CONTROL SYSTEM OF INVERTED PENDULUM

The FLC is a control system based on Fuzzy Set Theory, developed by Zadeh in 1965. Instead of relying on precise mathematical models, the FLC operates using linguistic control rules, closely mimicking human decision-making processes in uncertain and nonlinear environments.

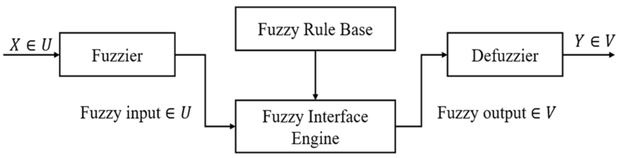
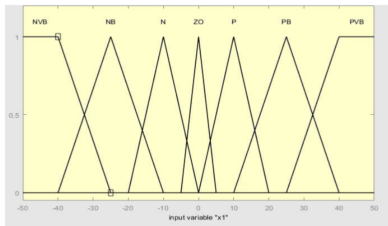


Fig. 2. Block diagram of the controller

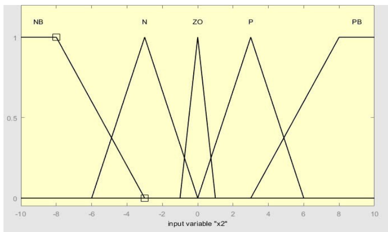
Fuzzy systems have been applied across various fields, including control, signal processing, communications, medicine, expert systems, and business, among others. However, most significant applications are concentrated in control-related problems. As illustrated in Fig. 2, fuzzy systems can be utilized as either open-loop or closed-loop controllers. As shown in Fig. 3, when a fuzzy system is used as an open-loop controller, the system typically establishes control parameters, and the system then operates according to these parameters. Conversely, when it is employed as a closed-loop controller, as depicted in Fig. 4, the fuzzy system continuously takes the output of the controlled system and applies control actions to the system being controlled.

FLC is designed to control systems that are difficult to model accurately or highly nonlinear systems, where traditional control methods Z(such as PID) prove to be ineffective. In the inverted pendulum system, the fuzzy controller uses the state variables of angle and angular acceleration to control the nonlinear system states.

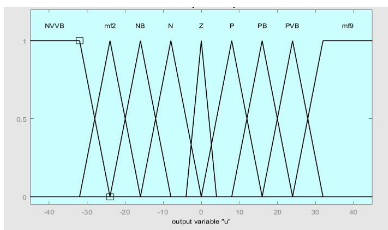
To control the system, with the state vector $(\theta/\dot{\theta})=(x_1\ x_2)$, we choose two state variables: $-40 \le x_1 \le 40$ and $-8 \le x_2 \le 8$ (Degree/Second). First, we construct the membership functions for x_1 with the following values: very large positive (PVB), large positive (PB), positive (P), zero (ZO), negative (N), large negative (NB), and very large negative (NVB):



a)



b)



c)

Fig. 3. a) Input membership function x1, b) Input membership function x2, c) Output membership function u

Next, we define the fuzzification method using the Singleton Fuzzification method, where the crisp inputs are mapped to fuzzy sets based on the membership functions. The inference system is based on the Mamdani method, which involves applying fuzzy rules to the fuzzified inputs and generating fuzzy outputs. Finally, the defuzzification method used is the Center Average, where the fuzzy outputs are converted into crisp control signals by calculating the weighted average of the output membership functions. This approach ensures that the fuzzy controller can effectively process the nonlinear system's states and produce appropriate control actions for the system:

In the Product Inference Engine (PIE), we use the algebraic product for all t-norm operators and the maximum operation for all s-norm operators. Therefore, the product inference engine can be represented as follows:

$$\mu_{B'}(y) = \text{M max l} = 1 \left[\sup x \in U \left(\mu_{A'}(x) \prod_{i=1}^{n} \mu_{A_i^l}(x_i)\mu_{B^l}(y)\right)\right] \quad (6)$$

This means that, with a fuzzy set A' in U, the product inference engine will generate a fuzzy set B' in V according to the expression above:

Table 1. Fuzzy control system of inverted pendulum (FLC)

x1	x2				
	PB	P	Z	N	NB
PVB	PVVB	PVVB	PVB	PB	P
PB	PVVB	PVB	PB	P	Z
P	PVB	PB	P	Z	N
Z	PB	P	Z	N	NB
N	P	Z	N	NB	NVB
NB	Z	N	NB	NVB	NVVB
NVB	N	NB	NVB	NVVB	NVVB

We use Mamdani's minimum implication method and the min operator for all t-norm operators, along with the max operator for all s-norm operators. Therefore, MMIE is represented as follows:

$$\mu_{B'}(y) = \text{M max l} = 1 \begin{bmatrix} \sup x \in U \left((\min \left(\mu_{A'}(x), \right.\right. \\ \left.\left. \mu_{A_1^l}(x_1), \ldots, \mu_{A_n^l}(x_n), \mu_{B^l}(y)\right)\right) \end{bmatrix} \quad (7)$$

4. QPSO ALGORITHM

When optimizing using PSO, particles in the search space are adjusted based on their velocity and position trajectories, influenced by both their personal experience and the global experience of the swarm [20]. However, PSO has some limitations, such as being prone to getting stuck in local minima and losing its ability to explore effectively in later stages of the optimization process [21].

To address this, the QPSO algorithm was introduced, combining quantum mechanics to enhance global convergence capabilities [22]. Instead of using velocity, QPSO relies on probability distribution functions to adjust the particle positions, creating a broader distribution in the search space and increasing the ability to explore unknown regions [15].

QPSO is an optimization algorithm developed based on the quantum mechanical principles derived from PSO (Particle Swarm Optimization). Compared to the PSO algorithm, QPSO has a distinct feature in that the particles do not have a fixed state during their motion, which is analogous to the indeterminate motion in quantum mechanics. All particles collectively form a swarm.

In the QPSO algorithm, the particle position update is described by:

$$X_{id}(t+1) = p_{id} \pm \beta \cdot |p_{id} - X_{id}(t)| \cdot \ln(1/u) \quad (8)$$

In which:

p_{id}: Local gravitational force.

β: Convergence rate control coefficient.

$X_{id}(t)$: The position of particle i in the d- dimensional space at iteration t.

$u \in (0,1)$: A random number.

p_{id} is calculated as follows:

$$p_{id} = \phi \cdot \text{pBest}_i + (1-\phi) \cdot \text{gBest} \quad (9)$$

In which:

$\phi \in (0,1)$: A random number.

pBest_i: The optimal position of particle i at each iteration.

gBest: The optimal global position of the population.

pBest_i and gBest is calculated as follows:

$$\text{pBest}_i(t) = \begin{cases} X_i(t) & , \text{if } f(X_i(t)) < f(\text{pBest}_i(t-1)) \\ \text{pBest}_i(t-1) & , \text{if } f(X_i(t)) \geq f(\text{pBest}_i(t-1)) \end{cases} \quad (10)$$

$$\text{gBest}(t) = \begin{cases} \text{pBest}_i(t) & , \text{if } f(\text{pBest}_i(t)) < f(\text{gBest}(t-1)) \\ \text{gBest}(t-1), & \text{if } f(\text{pBest}_i(t)) \geq f(\text{gBest}(t-1)) \end{cases} \quad (11)$$

In which:

$f(x)$: Fitness function.

A lower objective function value corresponds to a better solution.

To avoid falling into local extrema traps at the end of the iterative process, Sun Jun et al. introduced the concept of the optimal mean position mBest in each iteration, where the optimal position of all particles is quantified and averaged as follows:

$$\mathrm{mBest} = \frac{1}{m}\sum_{i=1}^{m} p_i(t) \qquad (12)$$

In which:

m: Number of particles in the population (particle count).

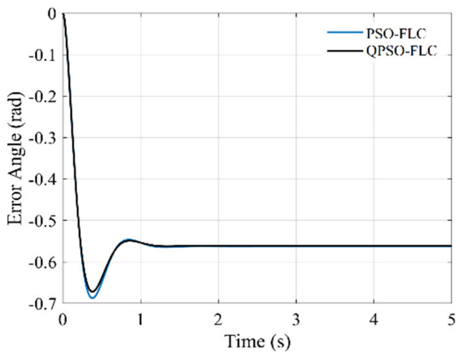
$p_i(t)$: Optimal position of particle i at time t.

Substituting into the above equation:

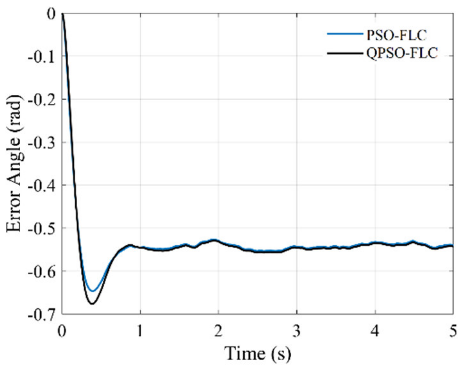
$$X_{id}(t+1) = p_{id} \pm \beta \cdot |\mathrm{mBest} - X_{id}(t)| \cdot \ln(1/u) \qquad (13)$$

5. RESULTS

The parameters of the inverted pendulum model are designed: m_1 = 0.08kg, m_2 = 0.098kg, L_1 = 0.16m, L_2 = 0.4m, I_1 = 0.0248kgm^2, I2 = 0.00386kgm^2, J_1 = 0.01Ns/m, J_2 = 0.01Ns/m, b_1 = 0.0136kg/m^2, b_2 = 0.00256kg/m^2.



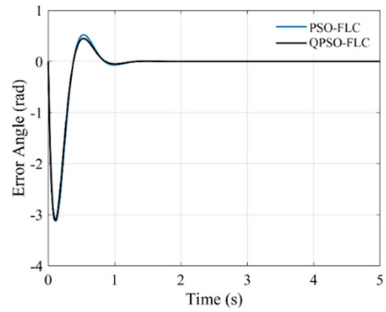
a) No noise



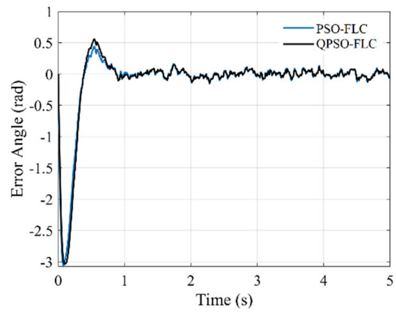
b) With noise

Fig. 4. Pendulum angle using the PSO-FLC and QPSO-FLC controllers

The simulation results are shown in the two figures above, with Fig. 4 representing the response of the pendulum angle and Fig. 5 representing the response of the arm angle when using the PSO-FLC and QPSO-FLC controllers. With the initial pendulum angle of -0.4rad, the system quickly stabilizes under 0.1rad after 2 seconds and approaches 0 rad after 5 seconds. In comparison, QPSO-FLC provides a more stable response with smaller oscillation amplitude than PSO-FLC. For the initial arm angle, the system stabilizes at 0 rad after 5 seconds, with a larger initial oscillation amplitude (-4rad). Again, QPSO-FLC shows better control performance in reducing oscillations compared to PSO-FLC. Thus, QPSO-FLC not only improves stability but also minimizes errors in the control system, outperforming PSO-FLC in both cases. The performance of the proposed controller is more clearly shown in Fig. 9. In which, the convergence in the optimization process and assessments based on quality criteria such as ISE, IAE, ITAE show that QPSO is lower than PSO.

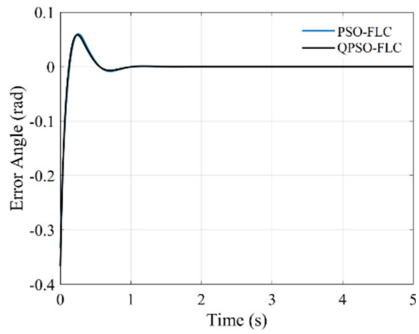


a) No noise

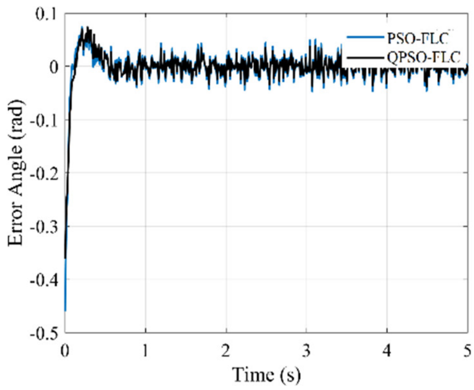


b) With noise

Fig. 5. Arm angle

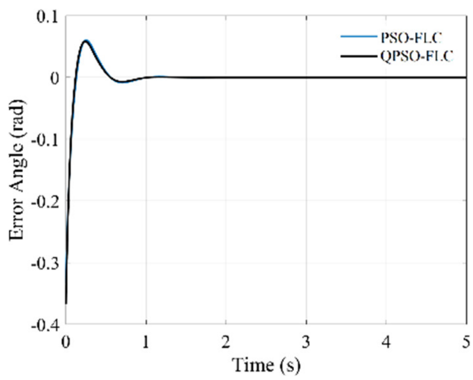


a) No noise

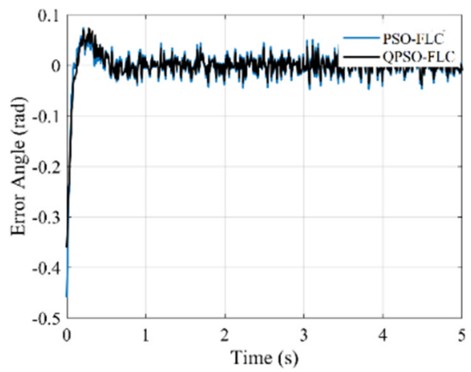


b) With noise

Fig. 6. Control signal

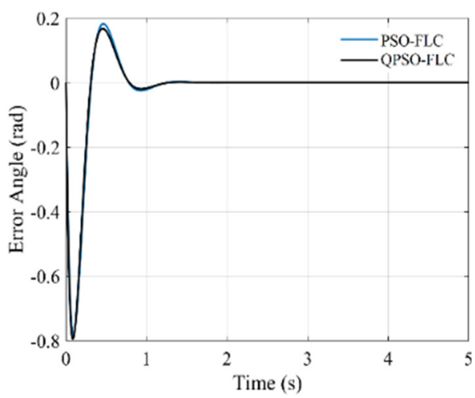


a) No noise

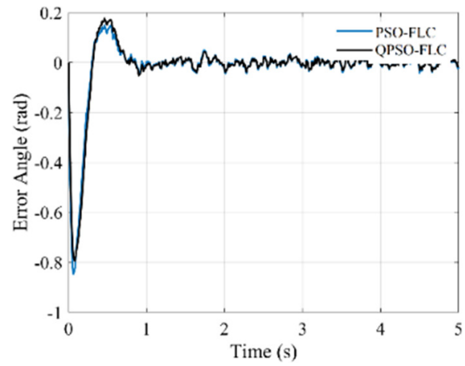


b) With noise

Fig. 7. Pendulum angle using the PSO-FLC and QPSO-FLC controllers

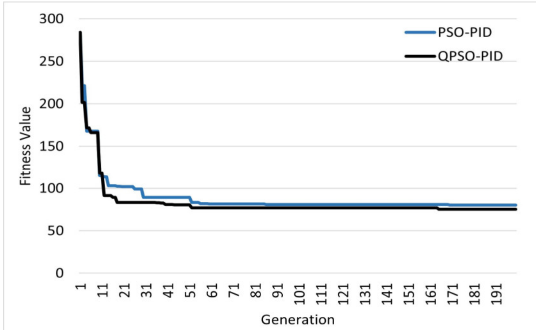


a) No noise



b) With noise

Fig. 8. Pendulum velocity using the PSO-FLC and QPSO-FLC controllers



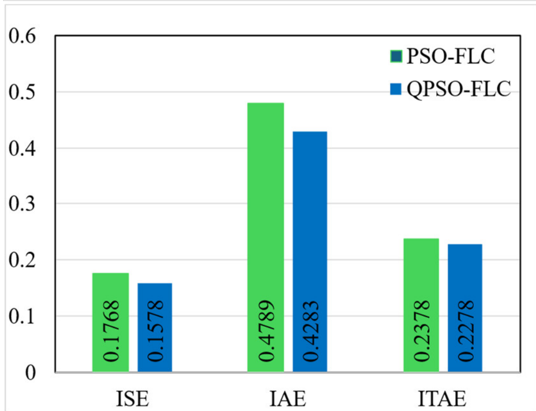


Fig. 9. Convergence graph (Left) and Evaluation of the controller according to different quality criteria for pendulum arm position error (Right)

Based on the system response graphs when using the PSO-FLC and QPSO-FLC controllers, several conclusions can be drawn. The QPSO-FLC controller outperforms the PSO-FLC in terms of reducing overshoot, shortening the response time, and increasing the system's stability. This demonstrates that QPSO-FLC is the more optimal method for this control problem.

5. CONCLUSION

The paper has successfully applied the balance control method for the reverse rotation pendulum by combining FLC and QPSO optimization algorithm. Compared to the traditional method using PSO, QPSO shows superior performance with faster convergence rate, better local manipular avoidance, and higher stability in noisy environments. Simulation and experimental results have confirmed that the QPSO-FLC controller significantly reduces oscillation, improves response time, and optimizes the quality index of the reverse rotation pendulum system. This opens up many future research directions, such as improving the QPSO algorithm, expanding the application to other mechatronic systems, or experimenting with adaptive control strategies to further improve the stability of the system.

REFERENCES

[1]. Baker G. J., Blackburn J. A., *The Pendulum: a Case Study in Physics*. Oxford University Press, New York, 2005.

[2]. M. Akhtaruzzaman, A. A. Shafie, "Modeling and control of a rotary inverted pendulum using various methods, comparative assessment and result analysis," in *Proceedings of the IEEE International Conference*, 2010.

[3]. Jadlovska S., Sarnovsky J., "An Extended Simulink Library for Modeling and Control of Inverted Pendula Systems," in *Proc. of the Int.Conf.Technical Computing Prague 2011*, Prague, Czech Republic, 2011.

[4]. Khalil S., *Inverted pendulum analysis, design and implementation*. Institute of the industrial Electronics Engineering. Karachi, Pakistan, 2003.

[5]. Reyes Sebastián, et al., "Takagi-Sugeno Fuzzy Control of the Cart-Inverted Pendulum System: Linear Matrix Inequalities Application," in *2024 IEEE International Conference on Automation/XXVI Congress of the Chilean Association of Automatic Control* (ICA-ACCA), IEEE, 2024.

[6]. Rani Mohd Rahairi, et al., "PID controller optimization for a rotational inverted pendulum using genetic algorithm," in *2011 Fourth International Conference on Modeling, Simulation and Applied Optimization*, IEEE, 2011.

[7]. Zadeh L. A., "Fuzzy sets," *Information and Control*, 8(3), 338-353, 1965.

[8]. Lee C. C., "Fuzzy logic in control systems: Fuzzy logic controller," *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2), 404-418, 1990

[9]. Cao S. G., Rees N. W., Feng G., "Analysis and design for a class of complex control systems using fuzzy logic," *Fuzzy Sets and Systems*, 122(1), 31-52, 2001.

[10]. Hazem Zied Ben, Mohammad Javad Fotuhi, Zafer Bingül, "A comparative study of the joint neuro-fuzzy friction models for a triple link rotary inverted pendulum," *IEEE Access*, 8 (2020): 49066-49078.

[11]. Gutarra Anthony, et al., "Fuzzy Controller Design for Rotary Inverted Pendulum System Using Genetic Algorithms," in *2022 IEEE Andescon*, IEEE, 2022.

[12]. Bejarbaneh Elham Yazdani, *Multi-optimization of PID controller parameters using stochastic search techniques for rotary inverted pendulum system*. Diss. Universiti Teknologi Malaysia, 2012.

[13]. Mai-Phuong Dao-Thi, et al., "Balancing a practical inverted pendulum model employing novel meta-heuristic optimization-based fuzzy logic controllers," *Int. J. Adv. Comput. Sci. Appl*, 13.4, 2022.

[14]. Sun Jun, et al., "Convergence analysis and improvements of quantum-behaved particle swarm optimization," *Information Sciences*, 193: 81-103, 2012.

[15]. Sun Jun, et al., "Quantum-behaved particle swarm optimization: analysis of individual particle behavior and parameter selection," *Evolutionary computation,* 20.3: 349-393, 2012.

[16]. Omkar S. N, et al., "Quantum behaved Particle Swarm Optimization (QPSO) for multi-objective design optimization of composite structures," *Expert Systems with Applications*, 36.8: 11312-11322, 2009.

[17]. David Reddipogu, Joshua Sunder, Vinodh Kumar Elumalai, "Hardware in the Loop Testing of Adaptive Inertia Weight PSO-Tuned LQR Applied to Vehicle Suspension Control," *Journal of Control Science and Engineering*, 1: 8873995, 2020.

[18]. Altinoz O. Tolga, A. Egemen Yilmaz, "Investigation of the optimal pid-like fuzzy logic controller for ball and beam system with improved quantum particle swarm optimization," *International Journal of Computational Intelligence and Applications,* 21.04: 2250025, 2022.

[19]. Karoly Lengyel, Ovidiu Stan, Liviu Miclea, "Seismic Model Parameter Optimization for Building Structures," *Sensors*, 20.7: 1980, 2020.

[20]. Kennedy James, Russell Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks,* 4, 1995.

[21]. Clerc Maurice, James Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE transactions on Evolutionary Computation*, 6.1: 58-73, 2002.

[22]. Sun Jun, Wenbo Xu, Bin Feng, "A global search strategy of quantum-behaved particle swarm optimization," in *IEEE Conference on Cybernetics and Intelligent Systems*, 1, 2004.