SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V12I4P125

Volume 12 Issue 4, 308-316, April 2025

Original Article

Improving Performance of Metaheuristic - Based Optimization Strategies: A Typical Application for PID-Family-Type Fuzzy Logic Controllers

Ngoc-Khoat Nguyen

Faculty of Control and Automation, Electric Power University, Hanoi, Vietnam.

Corresponding Author : khoatnn@epu.edu.vn

Received: 18 February 2025 Revised: 20 March 2025 Accepted: 19 April 2025 Published: 29 April 2025

Abstract - *This paper proposes a novel hybrid metaheuristic optimization technique to accelerate convergence speed. The proposed method effectively reduces convergence time and mitigates premature convergence by employing a rapid auxiliary optimization algorithm to generate an informed initial population. This innovative approach is implemented within two prominent bio-inspired algorithms, Particle Swarm Optimization (PSO) and the Bat Algorithm (BA). Specifically, the BA, known for its rapid convergence, is incorporated into the PSO's initialization stage, resulting in a substantial reduction in convergence duration. This efficient hybrid optimization strategy is applied to designing a PID-type Fuzzy Logic Controller (FLC), optimizing critical parameters, including scaling factors, membership functions, and fuzzy rules. The methodology is rigorously analyzed theoretically and through its application to the challenging nonlinear Ball and Beam (B&B) system. Extensive simulation studies encompassing a variety of operational scenarios demonstrate the superior control performance achieved over several conventional controllers, highlighting the practical applicability of the proposed hybrid control strategy.*

Keywords - *Metaheuristic-based optimization, Convergence time, Hybrid integration, Initialization, PID family–type FLC.*

1. Introduction

Optimizing the performance of control systems obviously needs more specific adjustments of controller parameters to conform to inherently competing performance objectives: rapid response, robust stability, and high steady-state accuracy. Determining optimal controller parameters represents a significant and fundamental challenge in control system design. The performance characteristics of the resulting closed-loop system are directly and critically dependent on the effectiveness of this tuning process.

This principle is universally applicable, encompassing traditional Proportional-Integral-Derivative (PID) regulators and sophisticated intelligent controllers, such as Fuzzy Logic Controllers (FLC) and artificial neural network ones, which demand precise scaling factor calibration [1-8].

Parameter tuning within the traditional PID control, a prevalent feedback control strategy, necessitates the adjustment of proportional, integral, and derivative gain coefficients. Tuning methodologies are principally categorized as empirical or model-based ones. Empirical techniques like the Ziegler-Nichols method rely on experimentally derived system response data. Conversely, model-based techniques leverage analytical plant models for parameter determination. The overarching objective of this process is to minimize tracking errors, attenuate overshoot, and ensure a stable and rapid system response to both setpoint variations and external disturbances. However, iterative manual tuning, often referred to as the "trial-and-error" method, presents significant challenges, including prolonged tuning times and suboptimal control performance. To mitigate these limitations, contemporary control systems frequently incorporate adaptive tuning algorithms. These algorithms enable real-time parameter optimization, thereby accommodating dynamic variations in operational conditions and enhancing overall system performance [2-5]. Similarly, the FLCs, particularly PID family-type fuzzy controllers, require the determination of input and output scaling factors analogous to the PID gains [8-10]. Consequently, efficient and systematic parameter optimization methodologies are essential. Various techniques have been employed for this purpose. For PID control systems that lack precise transfer function models, integrating optimization algorithms offers a viable solution. Figure 1 illustrates a schematic representation of a biomimetic optimization algorithm applied to PID controller tuning. Typically, each PID controller requires the optimization of three parameters: proportional gain (Kp), integral gain (Ki), and derivative gain (Kd), which serve as the optimization variables within the chosen algorithm [3].

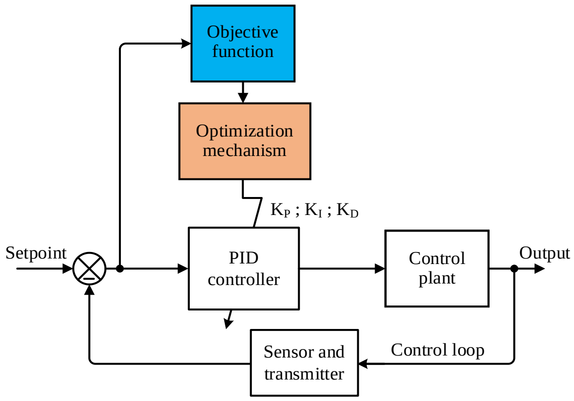

Fig. 1 Tuning a PID controller using optimization mechanisms

The FLCs are intelligent control systems, and they have been widely used in many applications of control [11, 15, 16] because they enable the design of controllers without requiring exact models of the systems. Often, the FLCs are used in conjunction with standard PID controllers [11] to tune the proportional (K_p), integral (K_i), and derivative (K_d) gains to improve the control action when there is a change in the model of the plant. In addition, biomimetic optimization algorithms are used today with FLCs, as shown in Figure 2 [3], to optimise the fuzzy controller's input and output scaling factors to obtain better performance.

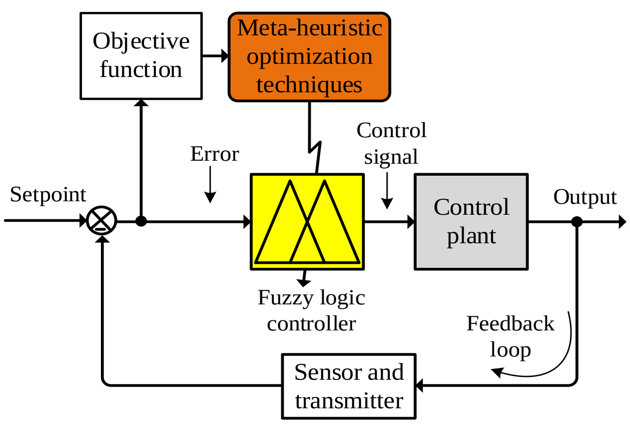

Fig. 2 Tuning a FLC using optimization methods

A significant challenge lies in the convergence rate of biomimetic optimization algorithms. These algorithms, particularly those exhibiting global optimization capabilities, often exhibit protracted convergence times. Research efforts must focus on improving existing algorithms to mitigate this issue and enhance the control performance of the aforementioned controllers. This study investigates a novel strategy involving the implementation of a rapid sub-optimization algorithm as an initialization phase for the global optimization algorithm. This hybrid approach is anticipated to accelerate convergence substantially. While fundamental optimization algorithms, such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), have been extensively studied and applied in domestic research [12-16], a substantial portion of these investigations primarily focus on basic algorithm implementation, neglecting in-depth analyses of convergence speed. Moreover, applying these optimization algorithms to FLC tuning has been largely limited to coefficient adjustments, overlooking other critical aspects of FLC design, such as membership function definition and fuzzy rule optimization. This paper proposes a comprehensive methodology to enhance the control quality of an FLC, especially FLCs based on the PID family. The remainder of this paper is structured as follows. Section 2 outlines the fundamental principles of the traditional PSO algorithm. Section 3 details a step-by-step design procedure for an enhanced PSO algorithm, accompanied by relevant analyses and explanations. Subsequently, Section 4 presents a case study illustrating the application of the proposed control methodology to the classic ball and beam balancing system. Finally, Section 5 offers concluding remarks and discusses potential avenues for future research.

2. Traditional PSO Algorithm

Particle Swarm Optimization (PSO), one of the most efficient metaheuristic optimization methods, operates following a mechanism inspired by the social behavior of bird or fish flocks searching for food. This algorithm is based on the interaction between inter-particles within the swarm for food navigation, and such a characteristic is different from other evolutionary optimization methods, such as the Genetic Algorithm (GA). Each particle within the PSO swarm represents a potential solution characterized by a position vector within the search domain. The particle's trajectory is dynamically adjusted based on two key metrics: the particle's personal best position (pbest), representing the optimal solution encountered by that individual, and the global best position (gbest), which can be considered the optimal solution discovered by the entire swarm. A typical PSO algorithm, with its flowchart presented in Figure 3, can be described in several main phases below:

Phase 1: Initialization: A population of particles is randomly distributed within the search space.

Phase 2: Calculate and evaluate the fitness function: The objective function is evaluated for each particle, yielding a fitness value. Each particle updates its pbest if its current fitness value surpasses its previous best. The gbest is concurrently updated if a particle discovers a superior solution compared to the current global best.

Phase 3: Position and Velocity Update: Each particle's position and velocity are updated according to its pbest, gbest, and algorithm-specific parameters, typically involving inertia and acceleration coefficients.

Phase 4: Termination Check: The algorithm terminates if a predefined stopping criterion is met, such as reaching a maximum number of iterations or achieving a satisfactory fitness convergence. Otherwise, the process returns to phase 2.

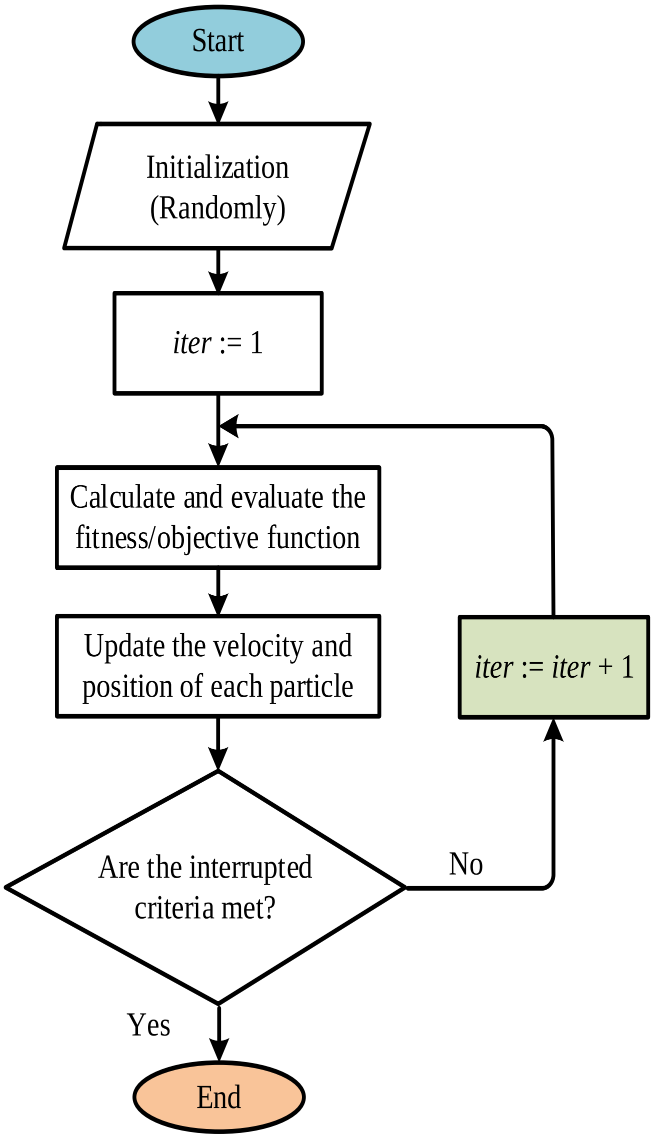

Fig. 3 The flowchart of a typical PSO algorithm

3. Enhanced PSO Algorithm

3.1. Evaluation of the Convergence of PSO

Building upon the established advantages of the PSO algorithm, specifically its algorithmic simplicity and ease of implementation, this study explores enhancements to reduce convergence time. A common challenge in optimization algorithms, including PSO, is premature convergence to local optima, particularly when the initial population or iteration count is inadequate. As illustrated in Figure 4, insufficient population diversity can preclude the algorithm from exploring regions containing either local or global optima, leading to suboptimal solutions. Additionally, the stochastic initialization of particle positions can contribute to prolonged convergence. While increasing population size or iteration count may mitigate these issues, it inevitably incurs a substantial computational overhead, extending the algorithm's execution time.

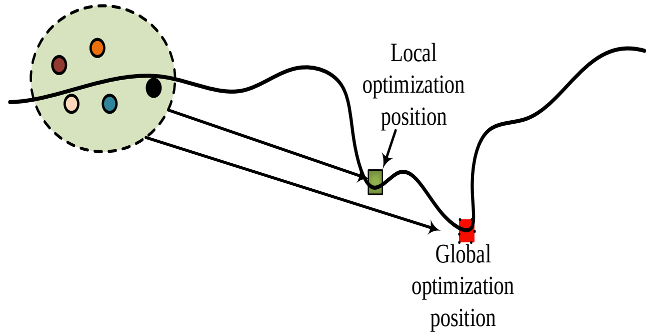

Fig. 4 Representation of convergence for a typical PSO

3.2. Improving the Convergence of a PSO Algorithm

Figure 5 conceptually depicts the proposed methodology for enhancing the convergence rate of the PSO algorithm. This approach assumes the presence of a local optimum within the search space. Convergence acceleration is achieved by biasing the initial population's distribution towards a known optimal point, which may be either a local or global optimum.

This initial optimal point is determined by applying a computationally efficient optimization algorithm significantly faster than standard PSO or, in certain contexts, estimated heuristically via iterative trial-and-error adjustments.

Implementation of this methodology necessitates modification of the random initialization step within the PSO algorithm's initialization phase, as illustrated in Figure 3. The procedural steps for executing the enhanced PSO algorithm are as follows:

Step 1: Employ a rapid optimization algorithm, utilizing a reduced number of initialization units (e.g., population members or individuals) to generate an acceptable set of near-optimal parameters. In this work, the Bat Algorithm (BA) is selected as a fast optimization mechanism. Drawing inspiration from how bats use echolocation to find their way and hunt, the BA offers a distinctive method for solving optimization problems in control systems. It simulates a group of virtual bats that change their flight characteristics – speed, sound frequency, and loudness – to search for the best solutions. Because the BA naturally balances a broad search with focused refinement and tends to find good solutions quickly, it shows great potential for tackling intricate control system design issues, like fine-tuning parameters and optimizing controller performance. Thus, this algorithm is embedded in the initial phase of the PSO mechanism, creating an enhanced optimization method.

Step 2: Incorporate the near-optimal parameter set obtained in Step 1 into the initialization parameters of the subsequent PSO algorithm.

Step 3: Execute the standard PSO algorithm, adhering to the fundamental procedural steps outlined in Section 2 (refer to Figure 3).

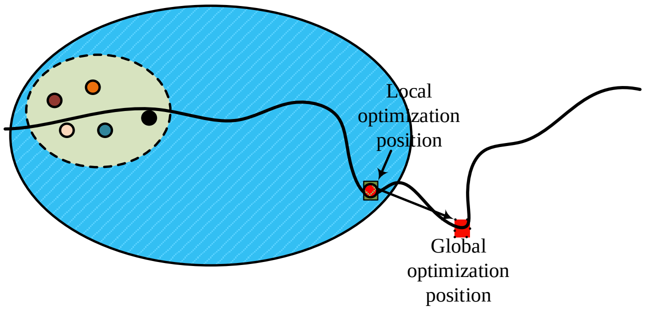

Fig. 5 Representation of convergence for the proposed PSO

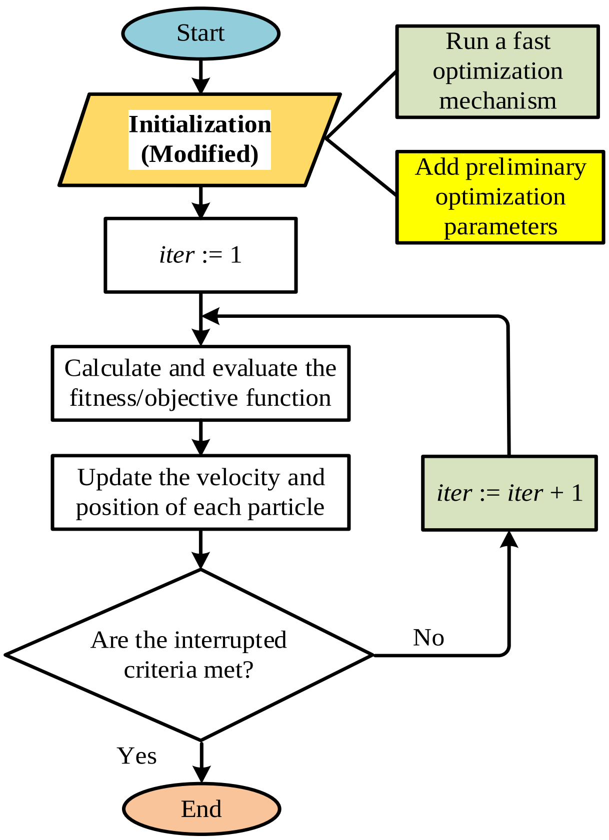

Fig. 6 The enhanced hybrid PSO algorithm flowchart

4. Results and Discussion

4.1. Ball and Beam System – A Typical Control System Example

In this section, the proposed control methodology will be applied to a typical control object: a Ball and Beam (B&B) system. This model is selected because it is considered a classical benchmark in control theory with dominant characteristics of nonlinearities and instabilities [11-18]. Figure 7 describes a simple physical model of such a B&B system. This model comprises four objects: a beam, a ball, a level arm, and a gear. The ball can roll along the length of the beam. The gear uses a servo motor, which can turn and is characterised by an angle denoted as θ (see Figure 7). The control objective is keeping the ball at several desired setpoints by swinging the beam from regulating the servo gear.

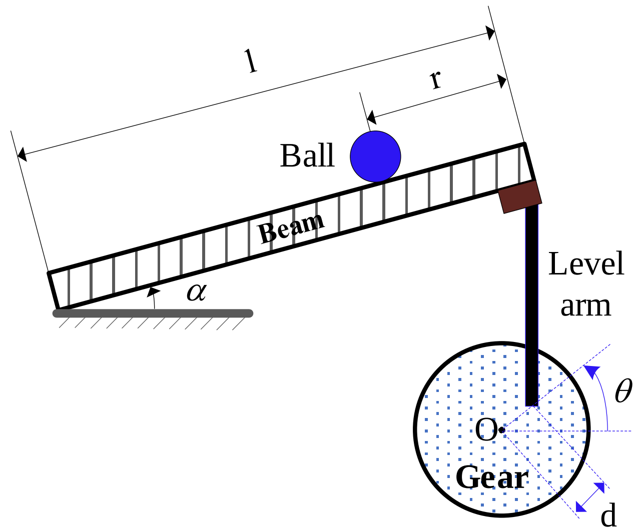

Fig. 7 A typical model of a B&B system

Consider Figure 7, with parameters given in the Appendix, neglecting the effect of the second derivative of α on the second derivative of *r*; the dynamics of such a B&B system are governed by the following:

(i) The Lagrangian equation, which is established for the motion of the ball following the Lagrangian mechanics as:

$$L = T - V = \left(\frac{J}{R^2} + m\right)\ddot{r} + mg\sin\alpha - mr\dot{\alpha}^2 = 0 \qquad (1)$$

Where:
L [J]: Lagrangian mechanics
T [J]: kinetic energy
V [J]: potential energy
l [m]: the beam's length
J [kg.m2]: the inertia of the ball
g [m/s2]: acceleration of gravity
d [m]: the offset of lever arm
R [m]: radius of the ball
m [kg]: the ball's mass
r [m]: real ball's position
α [rad]: the angle of the beam
θ [rad]: the angle of the servo gear

(ii) Linearizing (1) regarding a balancing state of the beam, a linear approximation of the system can be obtained below:

$$\left(\frac{J}{R^2} + m\right)\ddot{r} + mg\sin\alpha = -mg\alpha \qquad (2)$$

Where the beam angle α can be calculated in an approximated relationship with the angle of the gear θ as follows:

$$\alpha = \frac{d}{l}\theta \qquad (3)$$

(iii) The relationship between the position of the ball (controlled output) and the angle of the gear (input) can be deduced below:

$$\begin{cases} \left(\frac{J}{R^2} + m\right)\ddot{r} = -\frac{mgd}{l}\theta \\ r = \int_0^t \left[-\int_0^t \frac{mgd}{l\left(\frac{J}{R^2}+m\right)}\theta dt\right] dt \end{cases} \quad (4)$$

The above-approximated representation is a typical nonlinearity with double integration. Therefore, stable control is more challenging.

The main difficulty comes from instability, which calls for strong control strategies to bring disturbance action under control and to stabilize the system as well. While it is simply constructed physically, its dynamic behavior is complex; thus, it provides an excellent platform to test and compare different control algorithms.

4.2. Control Strategy for the B&B System

As mentioned above, the B&B system is suitable for testing the applicability of a robust control strategy. Here, three control methodologies have been embedded in the system, as shown in Figure 8, including:

(1) PD regulator,
(2) PID controller, and
(3) PD-based FLC.

The PID regulator has the following form:

$$u(t) = K_p.e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \quad (5)$$

where

- K_p, K_i and K_d are three tuning factors;
- $e(t)$ is the error signal, a difference between the setpoint and the output;
- $u(t)$ denotes the control signal which should be taken to an actuator in the control system.

PID controllers are valuable because they can be finely tuned by adjusting their proportional K_p, integral K_i, and derivative settings K_d. This adjustment allows engineers to strike the right compromise between reacting quickly to changes and maintaining a steady, oscillation-free system behaviour. In essence, it is possible to make the system respond promptly without becoming unstable or overly shaky.

For systems where oscillation reduction is paramount, PD controllers are a good fit. This type of conventional regulator is established when there is no integral part in the PID one (5). Their derivative component introduces damping, effectively minimizing overshoot and oscillations in the system's behaviour. This is particularly beneficial for mechanical systems like robotic arms and CNC machines, where accurate and stable motion is a key requirement.

Additionally, the PD controllers are beneficial when a rapid response to disturbances or setpoint changes is necessary. The proportional component acts immediately based on the present error, leading to a quicker reaction than if only integral control were used. Moreover, the derivative component anticipates future errors, allowing for proactive control action that further speeds up the initial response. Given these characteristics, a PD controller is theoretically well-suited for the inherently unstable B&B system.

Building upon this understanding, this research utilizes a PD-based Fuzzy Logic Control (FLC) structure. This architecture was specifically selected due to its proven effectiveness in stabilizing the naturally unstable dynamics of the ball and beam system. The whole control system is provided in Figure 9. Remember that every single PD-type FLC has three parts which need to be determined to ensure efficient operation as follows:

(a) Membership functions. Membership functions in a fuzzy controller map crisp inputs to fuzzy sets, quantifying the degree of membership within linguistic variables, thus enabling the processing of imprecise data.

(b) The fuzzy logic rules. The decision-making mechanism of a fuzzy controller relies on a rule set of "IF-THEN" statements. These rules, linking fuzzy inputs to outputs, facilitate control by mapping input membership to output actions, enabling the system to operate effectively with imprecise information.

(c) The processing factors. Scaling factors are employed in FLCs to transform input and output variables between their physical domains and the normalized domain of the fuzzy inference engine. Input scaling factors normalize sensor data to the fuzzy set range, and output scaling factors convert the resulting fuzzy control action back to physical units, thereby ensuring proper system interface and effective control execution.

The first and second parts have typically existed. In this section, following the empirical evaluation of various fuzzy logic models, a rule base derived from expert domain knowledge was selected, as documented in [2].

This rule set demonstrated superior performance in achieving the desired ball positioning and system stability. The *e*PSO algorithm and control methodology proposed in the previous section is modified, as depicted in Figure 10. After running the proposed hybrid optimization mechanism with parameters given in the Appendix, the simulation results obtained in MATLAB software are illustrated in Figures 11-17.

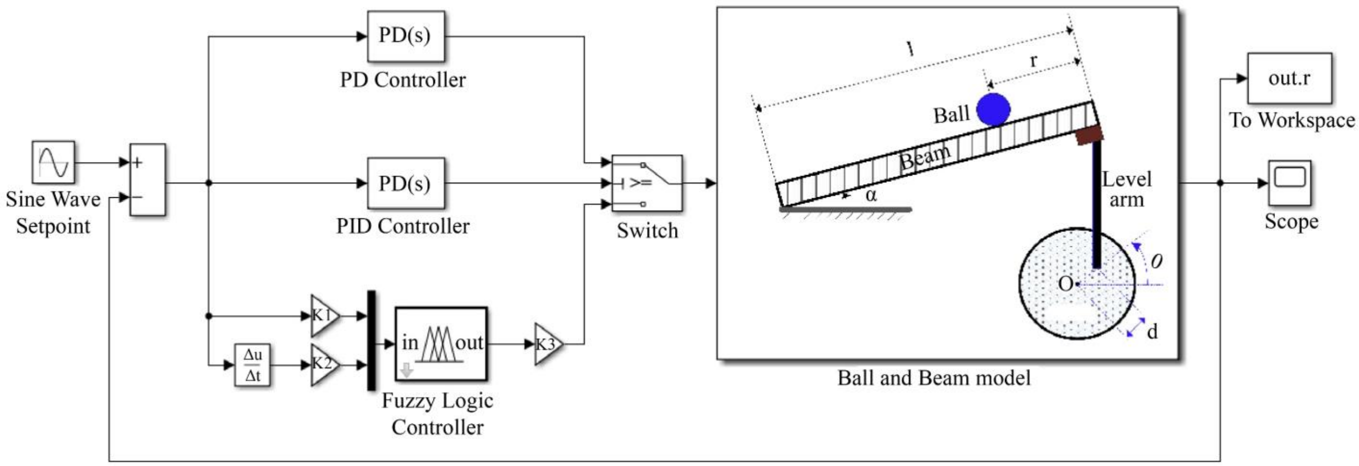

Fig. 8 Block control system built on MATLAB/Simulink environment

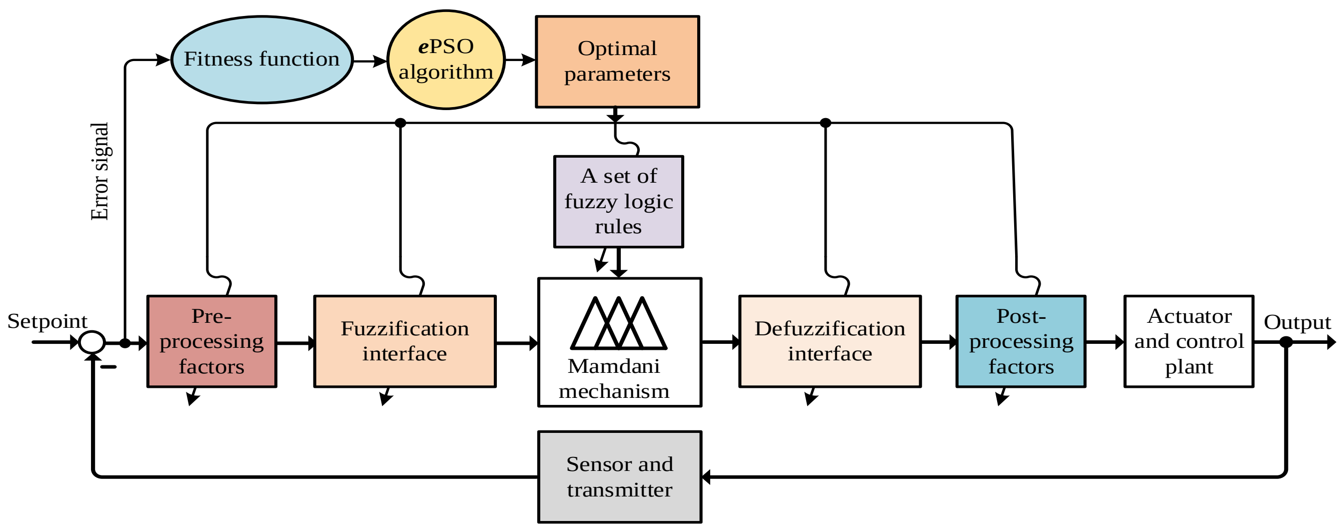

Fig. 9 Proposed hybrid FLC to a control plant

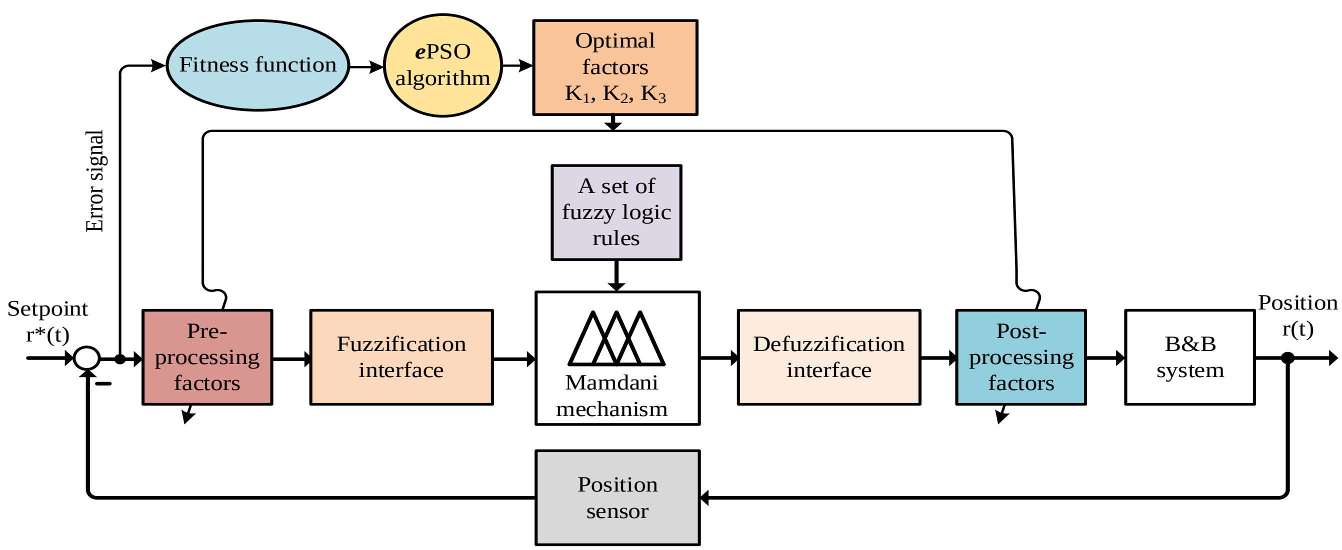

Fig. 10 The proposed FLC applied the *e*PSO algorithm for a B&B system

To implement the simulation process, two scenarios are applied regarding the setpoints of ball position. Figure 11 describes two setpoints: a sine wave and a complicated random one.

The sine wave with a unit amplitude and a frequency of 0.2 rad/s for a simulation time of 100s. The random number is employed as the second setpoint with a variance of 0.2 and a sample time of 50s for a simulation time of 500s.

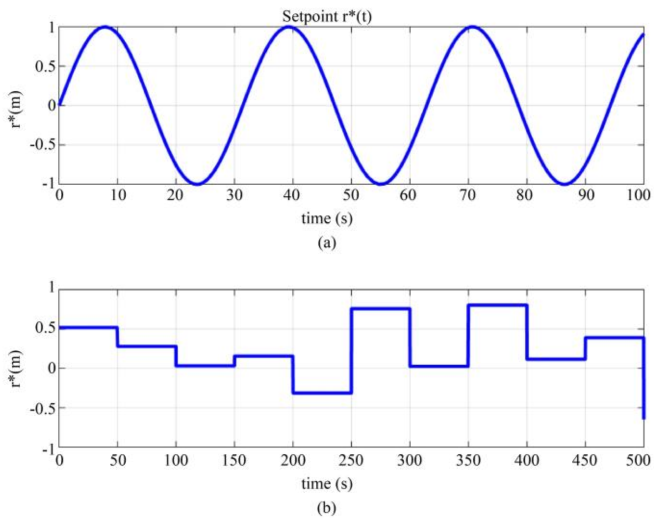

Fig. 11 Two scenarios of the setpoints applied to this control model a) Sine wave setpoint, b) Random setpoint

As illustrated in Figures 12-14, the traditional controllers (PD and PID) can force the output to be tracked at the setpoint with somewhat good performance in the first simulation case.

It is recognized in Figure 12 that the track of the PD controller is somewhat better than that of the PID regulator. This is explained by particular dynamic characteristics of the B&B system with a double integral as given in (4). Meanwhile, Figures 12-13 describe the dynamic response of the ball's position output.

Analysis of the output reveals tight setpoint tracking, validating the improved control efficacy of the proposed FLC. This is evidenced by a substantial decrease in deviation from the desired trajectory when contrasted with the performance of PD and PID controllers.

To evaluate in a more obvious representation, Figure 14 shows the ITAE (Integral of Time and Absolute Error) criterion, calculated in (6), which is considered to be one of the most effective control performances.

$$ITAE = \int_0^{\tau_{sim}} t * |e(t)|\, dt \qquad (6)$$

As a result of this control standard, the two conventional PD/PID controllers obtain ITAE values of nearly 600; meanwhile, the proposed hybrid FLC controller earns a much smaller number of 30. It means that the desired value of ITAE is less than 20 times the conventional ones. In the second random simulation scenario, a more complicated case, the proposed FLC continues attaining much better control quality, as shown in Figures 15-17. Obviously, the PD and PID controllers cause the results with bad performances, such as fluctuations and high overshoots.

In contrast, the proposed hybrid one, FLC, continues obtaining more exceptional performances, i.e., no overshoot and good steady time. Simultaneously, the measured value of the ITAE standard demonstrates a substantial decrease, several orders of magnitude lower than that observed with the two classical controllers, thereby validating the superior efficacy of the proposed FLC.

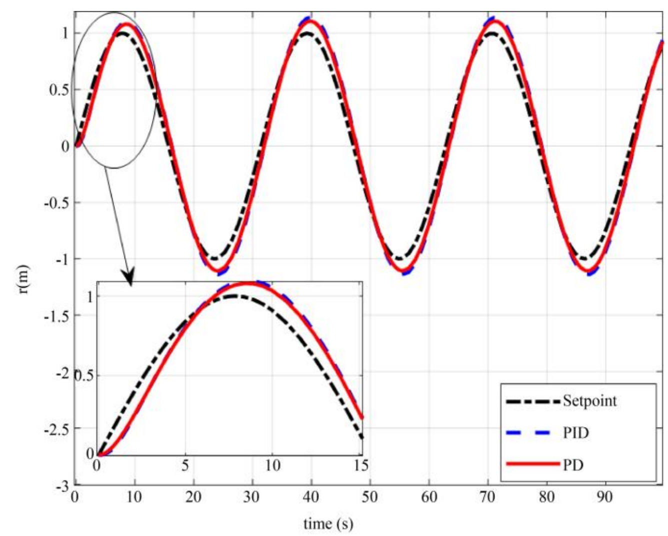

Fig. 12 Simulation result of the PD/PID in the first scenario

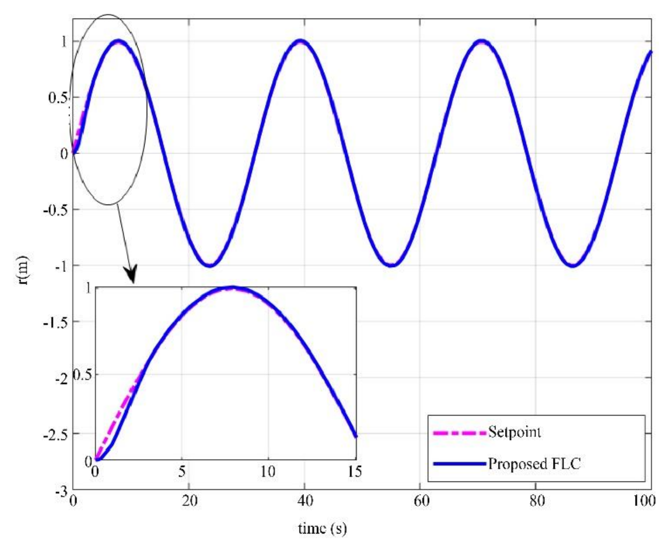

Fig. 13 Simulation result of the proposed FLC in the first scenario

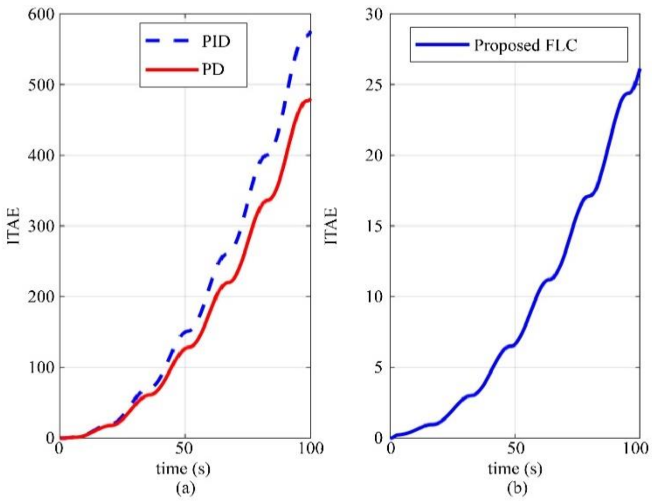

Fig. 14 Comparative ITAE of different controllers in the first scenario

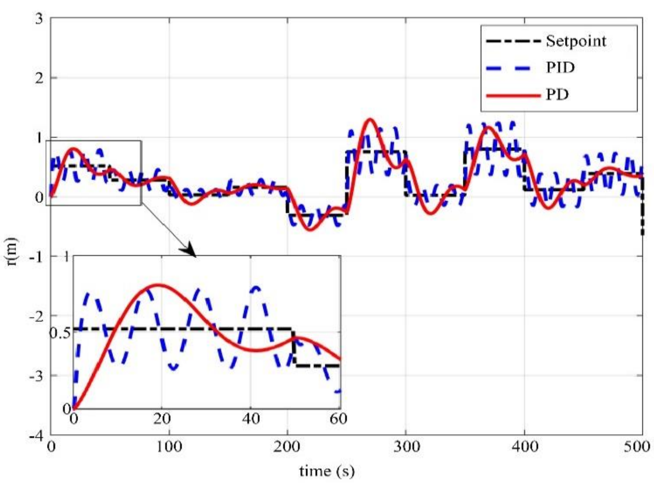

Fig. 15 Simulation result of the PD/PID in the second scenario

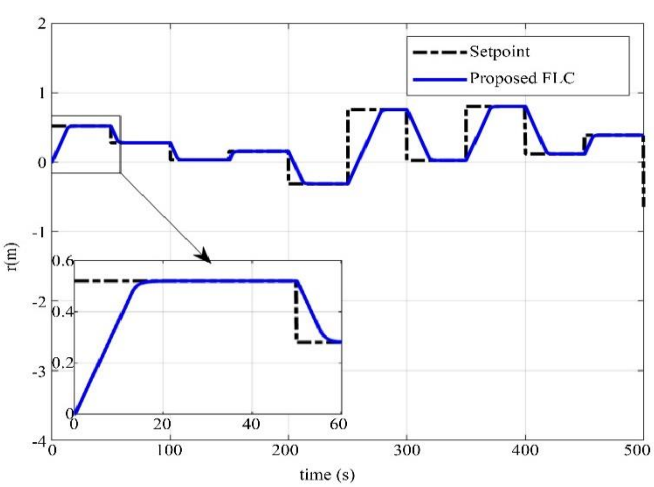

Fig. 16 Simulation result of the proposed FLC in the second scenario

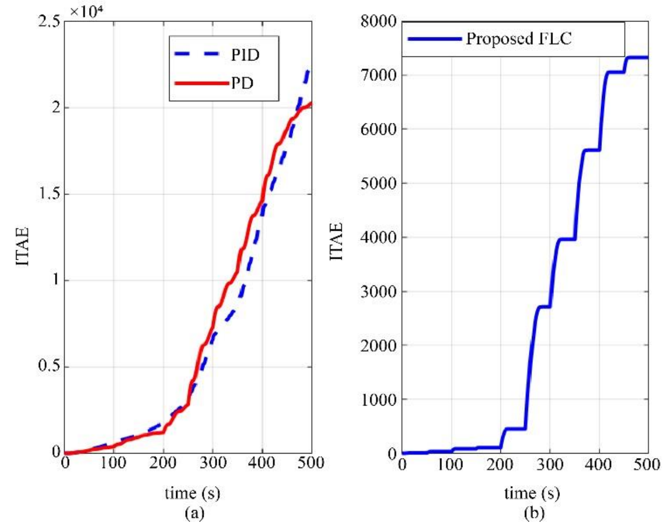

Fig. 17 Comparative ITAE of different controllers in the second simulation scenario

5. Conclusions and Future Works

This paper has dealt with improving metaheuristic method-based optimization algorithms, focusing on minimizing the convergence times.

This is implemented by modifying the initialization phase of the optimization algorithms. Some of the paper's major contributions can be deduced as follows:

(1) Design a procedure to reduce convergence time to enhance the implementation speed of the control strategy by applying the improved optimization methods.

(2) Apply the new finding above to a typical PSO algorithm to improve its control quality. A step-by-step given by flowchart for such an improved PSO mechanism has been proposed.

(3) An example of balancing a ball and beam system with nonlinearities applying the proposed *ePSO* has been conducted to verify the dominant control performances of the studied hybrid control strategy.

The next phases developed from this work will be the extension of improving ideas to other metaheuristic computation-based optimizations such as Genetic Algorithm (GA), Artificial Bee Colony (ABC), and Ant Colony Optimization (ACO).

In this future perspective, not only theory but also practical applications of the proposed control methodologies will be taken into consideration.

References

[1] Sergei S. Mikhalevich, Sergey A. Baydali, and Flavio Manenti, "Development of a Tunable Method for PID Controllers to Achieve the Desired Phase Margin," *Journal of Process Control*, vol. 25, pp. 28-34, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[2] J.C. Basilio, and S.R. Matos, "Design of PI and PID Controllers with Transient Performance Specification," *IEEE Transactions on Education*, vol. 45, no. 4, pp. 364-370, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[3] A. Jayachitra, and R. Vinodha, "Genetic Algorithm Based PID Controller Tuning Approach for Continuous Stirred Tank Reactor," *Advances in Artificial Intelligence*, vol. 2014, no. 1, pp. 1-8, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[4] Md. Manjurul Gani, Md. Saiful Islam, and Muhammad Ahsan Ullah, "Optimal PID Tuning for Controlling the Temperature of Electric Furnace by Genetic Algorithm," *SN Applied Sciences*, vol. 1, no. 8, pp. 1-8, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[5] Jun Zhao, Tianpeng Li, and Jixin Qian, "Application of Particle Swarm Optimization Algorithm on Robust PID Controller Tuning," *In Advances in Natural Computation, First International Conference*, Changsha, China, pp. 948-957, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[6] Xiaoyong Huang et al., "Application of an Improved PSO Algorithm to Control Parameters Auto-tuning for Bi-axial Servo Feed System," *Procedia CIRP*, vol. 41, pp. 806-811, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[7] Reshu Chaudhary, and Hema Banati, "Study of Population Partitioning Techniques on Efficiency of Swarm Algorithms," *Swarm and Evolutionary Computation*, vol. 55, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Jeydson Lopes da Silva, *Fuzzy Logic Control with PSO Tuning*, In Fuzzy Systems - Theory and Applications, Intech Open, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Huynh Thai Hoang, *Intelligent Control Systems, National University, HCM City University of Technology*, 2012. [Publisher Link]

[10] Onur Silahtar et al, "Beam and Ball Plant System Controlling Using Intuitionistic Fuzzy Control," *In Intelligence Science III, 4th IFIP TC 12 International Conference*, ICIS 2020, Durgapur, India, pp. 255-262, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Reza Moezzi, Trieu Minh Vu, and Mart Tamre, "Fuzzy Logic Control for a Ball and Beam System," *International Journal of Innovative Technology and Interdisciplinary Sciences*, vol. 1, no. 1, pp. 39-48, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[12] Mustafa Saad, and Mohammed Khalfallah, "Design and Implementation of an Embedded Ball-beam Controller Using PID Algorithm," *Universal Journal of Control and Automation*, vol. 5, no. 4, pp. 63-70, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[13] Ritu Raj, "PID-Based Optimized Fuzzy Controller Design for Ball and Beam System," *IEEE 5th International Conference on Recent Trends in Computer Science and Technology*, Jamshedpur, India, pp. 245-249, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] N. Muškinja, "Design, Development and Control of a Ball-on-Beam Control System Ising Industrial Equipment," *IEEE 46th MIPRO ICT and Electronics Convention*, Opatija, Croatia, pp.1555-1560, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Karan Sehgal, and Harsh, "Modelling and Control of Dynamical Ball and Beam System Using SA Tuned PIDA and PIAD Controllers," *IEEE International Conference on Electronics, Computing and Communication Technologies*, Bangalore, India, pp.1-6, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] Snigdha Chaturvedi, Narendra Kumar, and Rajesh Kumar, "Two Feedback PID Controllers Tuned with Teaching-Learning-Based Optimization Algorithm for Ball and Beam System," *IETE Journal of Research*, vol. 70, no. 7, pp. 6340-6349, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] Muhammad Ridho Rosa, Muhammad Zakiyullah Romdlony, and Bambang Riyanto Trilaksono, "The Ball and Beam System: Cascaded LQR-FLC Design and Implementation," *International Journal of Control, Automation and Systems*, vol.21, no.1, pp.201-207, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Neha Khanduja, and Bharat Bhushan, "Hybrid State of Matter Search Algorithm and its Application to PID Controller Design for Position Control of Ball Balancer System," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol.46, no. 3, pp.785-804, 2022. [CrossRef] [Google Scholar] [Publisher Link]

Appendix: Simulation parameters

B&B system

m = 0.0027 kg; r = 0.02 m; d = 0.0508 m; g = 9.81 m/s2; l = 1.0 m; J = 6.207*10^{-7} kg*m2

PSO algorithm

N = 12; npar = 3; c1 = 1.45; w = 0.65; max_iteration = 15; Lb = [0.1 0.1 0.1]; Ub = [5 5 5];

BA algorithm

Dim = 3; max_iter_BA = 3; N_BA = 12; Lb = [0.1 0.1 0.1]; Ub = [5 5 5]; Fmax = 3; Fmin = 0; r0 = 0.1%.