Synthesis of parameter adaptive PSO-PIDA controller with swarm optimization algorithm for hydraulic jack drive system

Nguyen Quang Binh*

Institute of Automation, Academy of Military Science and Technology, 89 Ly Nam De, Hoan Kiem, Hanoi, Vietnam.

*Corresponding author: qbinhk37@gmail.com

Received 6 Oct. 2025; Revised 11 Nov. 2025; Accepted 10 Dec. 2025; Published 25 Dec. 2025.

DOI: https://doi.org/10.54939/1859-1043.j.mst.108.2025.40-48

ABSTRACT

This paper presents a PSO-PIDA controller design for an electro-hydraulic actuation system driven by a proportional valve and supplied by an independent constant-pressure hydraulic source. The mathematical model of the actuator is derived from physical principles, incorporating valve flow dynamics, fluid compressibility, and piston motion to obtain a third-order transfer function. The Particle Swarm Optimization (PSO) algorithm is employed to determine the optimal parameters of the PIDA controller, aiming to minimize a composite time-domain performance index. Simulation results demonstrate that the proposed PSO-PIDA controller significantly improves transient response, reduces overshoot, and enhances pressure stability compared with Ziegler-Nichols-PID and PSO-PID controllers. The large parameter values obtained from PSO tuning are interpreted in terms of acceleration compensation, which increases control effort but yields faster settling and higher precision. The study confirms that PSO-PIDA control offers a practical and effective approach for high-performance electro-hydraulic systems and provides a clear foundation for future experimental validation.

Keywords: Electro-hydraulic actuator; PSO optimization; PIDA controller; Proportional valve; Nonlinear control.

1. INTRODUCTION

Electro-hydraulic actuation systems are widely used in aerospace, construction, and defense applications due to their high power density, fast response, and precise control capability. However, the nonlinear characteristics caused by fluid compressibility, valve dead zones, and internal leakage make controller design a challenging task [12-14]. Classical proportional-integral-derivative (PID) controllers remain dominant in industrial applications because of their simplicity and robustness [7]. Yet, conventional tuning rules, such as the Ziegler-Nichols (ZN) method, are often insufficient for systems with high nonlinearity or varying dynamics, leading to overshoot, oscillation, or slow convergence. To overcome these limitations, advanced control structures and intelligent optimization algorithms have been introduced. Among them, the Proportional-Integral-Derivative-Acceleration (PIDA) controller extends the conventional PID by adding an acceleration feedback term, which improves damping and transient precision for higher-order systems [1, 2]. Early studies demonstrated the analytical design of PIDA controllers for third-order plants, providing better phase margin and tracking accuracy compared with conventional PID schemes. However, analytical tuning approaches require accurate system parameters, which are often unavailable or uncertain in electro-hydraulic systems.

In parallel, metaheuristic optimization techniques, including Genetic Algorithms (GA), Differential Evolution (DE), and Particle Swarm Optimization (PSO), have proven effective for automatic controller tuning without requiring gradient information [3-6]. These algorithms can efficiently explore large search spaces and identify near-optimal parameters that balance overshoot, rise time, and energy consumption. In particular, PSO, inspired by the collective behavior of bird flocks, offers a simple structure and fast convergence with fewer control parameters than GA [6, 8]. Recent studies have successfully applied PSO to enhance PID control performance in nonlinear and electro-hydraulic systems. Chang and Shih [5] introduced an improved PSO approach for nonlinear systems, while Wang and Li [9] and Xiao et al. [14] demonstrated that PSO-tuned PID controllers can significantly improve transient response and robustness in hydraulic servo applications. Unnithan et al. [10] applied PSO to a semi-active electro-hydraulic damper, highlighting its potential for real-time control. Similarly, Guo et al. [12] and Skarpetis et al. [11] confirmed the efficiency of PSO-based methods for hydraulic actuators under parameter uncertainties. In Vietnam, Nguyen et al. [4] successfully implemented a genetic optimization algorithm for a vehicle-mounted hydraulic balancing system, indicating growing local interest in intelligent control for electro-hydraulic applications.

Motivated by these findings, this paper proposes a PSO-PIDA controller for an electro-hydraulic actuator system driven by a proportional valve under constant pressure supply. The system model is derived from physical principles, including valve flow dynamics, oil compressibility, and piston motion, resulting in a third-order transfer function. The PSO algorithm is utilized to automatically optimize the PIDA controller gains (K_p, K_i, K_d, K_a) according to a composite time-domain performance index. The proposed controller is then compared with ZN-PID and PSO-PID controllers to evaluate improvements in displacement response, control effort, and pressure stability.

2. SWARM OPTIMIZATION ALGORITHM

2.1. Introduction to the swarm optimization algorithm

PSO is a metaheuristic optimization technique inspired by the collective behavior of biological swarms. Each particle adjusts its position in the search space based on its best experience and that of the entire swarm. The algorithm iteratively updates velocity and position as in (1) - (3), ensuring convergence to the global optimum (figure 1).

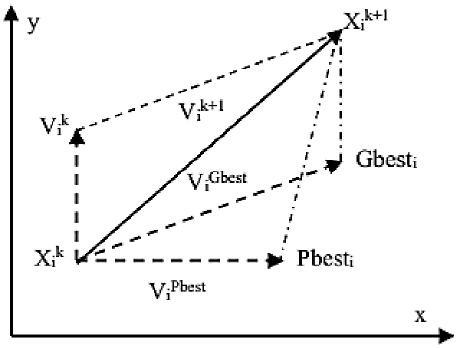


Figure 1. *Describes PSO's velocity and position updates in space.*

The process of updating the velocity and position of the elements is performed according to the following formulas:

$$v_{i,m}^{t+1} = wv_{i,m}^{t} + c_1\text{rand}(0,1)\left(P_{best_{i,m}} - x_{i,m}^{t}\right) + c_2\text{rand}(0,1)\left(G_{best_{i,m}} - x_{i,m}^{t}\right) \quad (1)$$

$$x_{i,m}^{t+1} = x_{i,m}^{t} + v_{i,m}^{t+1};\ i = 1,2 \dots, N;\ m = 1,2 \dots, D \quad (2)$$

Where: N is the number of swarms; D is the population size; t is the number of iterations; $v_{i,m}^{t}$ is the velocity of the i-th element in loop *t*; w is the inertial weight coefficient; c_1, c_2 are the acceleration coefficients of each element and the whole swarm; rand(0,1) is the random function with values in the range (0,1); $x_{i,m}^{t}$ is the position of the i-th element in loop *t*; $P_{best_{i,m}}$ is the best position of the i-th individual; $G_{best_{i,m}}$ is the best position of the i-th individual in the population.

To balance the global and local search capabilities of the PSO algorithm, use equation (3) to adjust the inertial weight coefficient [9].

$$w = w_{max} - \frac{w_{max} - w_{min}}{T_{max}} t \quad (3)$$

2.2. Design of PID, PIDA controllers with swarm algorithm

The PIDA controller (figure 2) is useful in improving phase margin, steady-state accuracy, and stability for higher-order systems. Its transfer function is shown below:

$$C(s) = K_p + \frac{K_i}{s} + K_d s + K_a s^2 \quad (4)$$

Or (4) rewrite in the form:

$$C(s) = K \frac{(s+a)(s+b)(s+z)}{s} \quad (5)$$

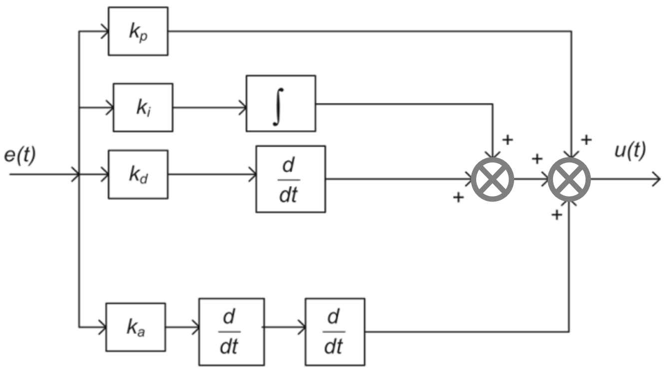


Figure 2. *Structure of PIDA controller.*

We have the following closed-loop transfer function system :

$$T(s) = \frac{G(s)C(s)}{1 + G(s)C(s)} \quad (6)$$

Then $1 + G(s)C(s)$ is the characteristic equation reflecting the quality of the system response process.

According to [1], the process of designing a PIDA controller for a third-order system using the root locus method includes the following steps:

Step 1: Determine ξw_n, the dominant solution, from the characteristics of the settling time $T_s = \frac{4}{\xi w_n}$. Thus, the dominant pair of complex conjugate poles is $s = q$ và $\hat{q}$, with $q = -\xi w_n + j w_n \sqrt{1 - \xi^2}$.

Step 2: Determine the attenuation coefficient ξ according to the characteristics of the overshoot $PO = 100 \cdot \exp\left(-\frac{\xi \pi}{\sqrt{1-\xi^2}}\right)$.

Step 3: Choose a real pole R with a value such that $R = Re\{q\} \leq -\xi w_n$.

Step 4: Choose a real pole r with a value such that $r \ll -\xi w_n$

Step 5: Write the characteristic equation :

$$1 + G(s)C(s) = 0 \text{ and } (s + r)(s + R)(s + q)(s + \hat{q}) = 0$$

Step 6: Find the unknowns by identifying the coefficients of the two equations in step 5.

Step 7: Draw a graph to observe the system's output response. If the overshoot is not satisfactory, increase the coefficient K.

Thus, to design a PID or PIDA controller by mathematical analysis method mainly depends on the designer's experience, but the solution has not yet achieved optimality in the root locus space.

Figure 3 presents the block diagram of the PID, PIDA controller design based on the PSO algorithm. The two controllers, PSO-PID, PSO-PIDA, differ only in the population size. This process continues until the objective function J is achieved in the last iteration.

The objective function J is the function used to evaluate the solution of the problem, where the error $e(t)$ between the set signal and the feedback signal quickly converges to 0 in a certain period of time. According to the ISE integration criterion (7), we have :

$$J = \int_0^t e^2(\tau)d\tau \tag{7}$$

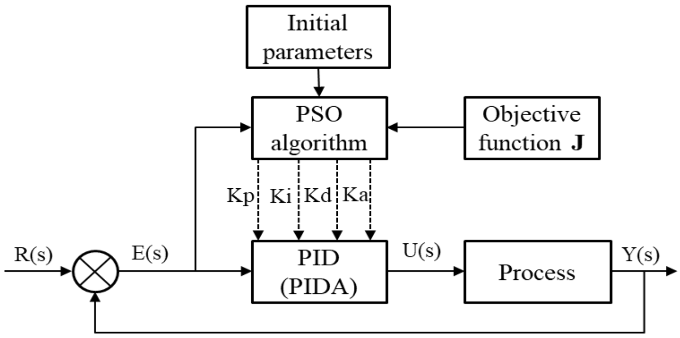


Figure 3. *PID, PIDA controller design diagram based on PSO algorithm.*

The PSO algorithm to optimize the PIDA controller parameters is performed as follows:

Step 1: Initialize population size *D*, swarm number *N*, inertial weight coefficient limit, and acceleration coefficient c_1, c_2, limit the search space for K_p, K_i, K_d, K_a (U_b - Upper bound, L_b - Lower bound), maximum number of iterations T_{max}. Initialize each individual in the population with a random position value and, velocity value P_{best}, G_{best}.

Step 2: Start loop, calculate the value of the objective function.

Step 3: Evaluate the position of the i-th individual at iteration k, i.e X_i^k according to the objective function value. Then update P_{best}, G_{best}.

Step 4: Update the inertial weight coefficient according to equation (3).

Step 5: Update the velocity and position values for each individual according to equations (1), (2). Check the search space limits for position and speed.

Step 6: Repeat step 2 until enough iterations are completed. Then G_{best} is found, and corresponding to it is the best position (i.e, the optimal set of parameters K_p, K_i, K_d, K_a optimal) of the individual in the population.

3. SIMULATION AND DISCUSSION

3.1. System model

The control object includes the electric valve is supplied with voltage $u(t)$ to control the displacement $x(t)$ for the jack cylinder with piston area A. The hydraulic source (pump) always provides sufficient source pressure; the difference pressure between the 2 valves' gates is $p(t)$.

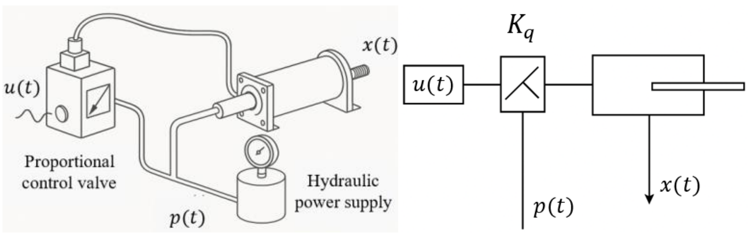


Figure 4. *Electro-hydraulic actuation systems (EHAS).*

Piston force (according to Newton):

$$M\ddot{x}(t) + B\dot{x}(t) + K_x x(t) = Ap(t) - F_{load}(t) \quad (8)$$

With: K_x is the mechanical stiffness; F_{load} is the external force. The equivalent viscous friction is B. The piston and load masses are M. The valve has a linear flow coefficient K_q and a pressure-damping coefficient K_c (describing the flow reduction due to back pressure).

Volumetric continuity equation (compression and leakage):

$$\frac{V_t}{\beta_e}\dot{p}(t) + C_t p(t) + A\dot{x}(t) = q(t) \quad (9)$$

Where, V_t is the volume of the oil chamber; β_e is the effective volume modulus; C_t is the leakage coefficient. Linear valve characteristics (linearization around operating point):

$$q(t) = K_q u(t) - K_c p(t) \quad (10)$$

With K_q is the valve gain coefficient - flow when $u = 1V$; K_c is the pressure feedback coefficient to flow.

Take the Laplace transform without an external force F_{load}. From (8):

$$(Ms^2 + Bs + K_x)X(s) = A\,P(s) \quad (11)$$

From (9) and (10):

$$\left(\frac{V_t}{\beta_e}s + C_t + K_c\right)P(s) + AsX(s) = K_q U(s) \quad (12)$$

With setting $D(s) = \frac{V_t}{\beta_e}s + C_t + K_c \Rightarrow D(s)P(s) + AsX(s) = K_q U(s)$

Take $P(s)$ from (12) and combine with (11):

$$(Ms^2 + Bs + K_x)X(s) = A\frac{K_q U(s) - AsX(s)}{D(s)} \quad (13)$$

Arrange to withdraw $X(s)/U(s)$:

$$X(s)\left(Ms^2 + Bs + K_x + \frac{A^2 s}{D(s)}\right) = \frac{AK_q}{D(s)}U(s) \quad (14)$$

Multiply both sides of (14) by $D(s)$:

$$X(s)[D(s)(Ms^2 + Bs + K_x) + A^2 s] = AK_q\,U(s) \quad (15)$$

$$\Rightarrow \frac{X(s)}{U(s)} = \frac{AK_q}{D(s)\,(Ms^2 + Bs + K_x) + A^2 s} \quad (16)$$

The formula (16) is the 3rd order transfer function. The convention of p.u. (per unit) is the control input normalized to $U_{rated} = 10V$ (displayed by $u(t)/U_{rated}$). Pressure is globally normalized to the maximum value obtained to facilitate dynamic comparison between controllers. The simulation is conducted with $M = 1.0\,kg$; $B = 1.4\,N \cdot s/m$; $\alpha = V_t/\beta_e = 0.2$; $C_{total} = 1.12$; $A = \sqrt{0.432}\,m^2$; $K_q = 0.456\,m^3/(s \cdot V)$. Applying the reduced dynamics model (17), the system model has been achieved in (18).

$$\alpha M\dddot{x} + (\alpha B + C_{total}M)\ddot{x} + (C_{total}B + A^2)\dot{x} = AK_q\,u(t) \quad (17)$$

Thus, the foot jack drive mechanism achieved is similar to the transfer function in [5]:

$$G(s) = \frac{0.3}{0.2s^3 + 1.4s^2 + 2s} \quad (18)$$

The process of running the PSO algorithm is simulated on MATLAB software with the signal set as a unit step function. In addition, the parameters for the PSO algorithm are taken from table 1.

It can be seen that the foot jack drive mechanism is a high-order system (order 3) in which the PIDA controller will give better control quality than the conventional PID.

***Table 1.** Parameters for PSO algorithm.*

N	D	T_{max}	c_1, c_2	w_{max}	w_{min}	U_b	L_b
20	4	30	2,05	1	0,1	1000	0

The PSO algorithm works based on the interaction between individuals in the herd during the foraging process, so the search space is infinite. To implement the algorithm, it is necessary to limit the search space and only find the optimal solution in that space, which means (L_b, U_b) = (0, 1000). The choice of U_b is based on the analysis results using Jung's method [2]. Thus, the coefficients K_p, K_i, K_d, K_a will always be within the limit range (L_b, U_b).

3.2. Model evaluation

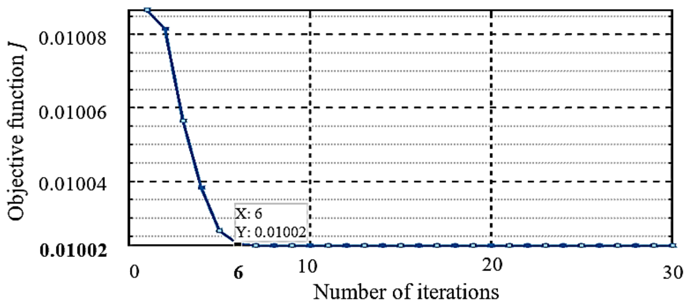


***Figure 5.** The process of evaluating the objective function J.*

Figure 5 shows that after 30 iterations, the objective function defined by equation (7) has a decreasing value and gradually converges at the beginning of the 6th iteration, when $J_{min} = 0{,}01002$. Table 2 shows that the parameter set of the PIDA controller is considered optimal with (K_p, K_i, K_d, K_a) = (1000; 0; 1000; 302,05). It can be seen that the coefficient $K_p = K_d = U_b = 1000$ proves that they are always in the search space (L_b, U_b) and the coefficient $K_i = 0$ is found by metaheuristic algorithms because the transfer function of the actuator has an ideal integral, so the closed-loop system always keeps up with the change of the input signal as a unit step function with steady-state error $e_{xl} = 0$.

***Table 2.** Coefficients of PID and PIDA controllers.*

Coefficients	ZN-PID	GA-PID	PSO-PID	PSO-PIDA
K_p	28	31.7	33.8	1000
K_i	28	0.44	0	0
K_d	7	27.0	35.1	1000
K_a	-	-	-	302.05

Based on table 3 and figure 6, with the same controller, the metaheuristic algorithms will give relatively similar results; the system response quality criteria are not significantly different, such as the GA-PID settling time is 1.39 seconds and PSO-PID is 1.38 seconds, much better and more optimal than the ZN-PID method (5.8 seconds).

***Table 3.** Comparison of system response quality.*

Target \ Controller	ZN-PID [4]	GA-PID [4]	PSO-PID	PSO-PIDA
Settling time (s)	5.8	1.39	1.38	0.0096
Rise time (s)	0.419	0.253	0.247	0.005
Overshoot (%)	59.4	13.8	14.17	0

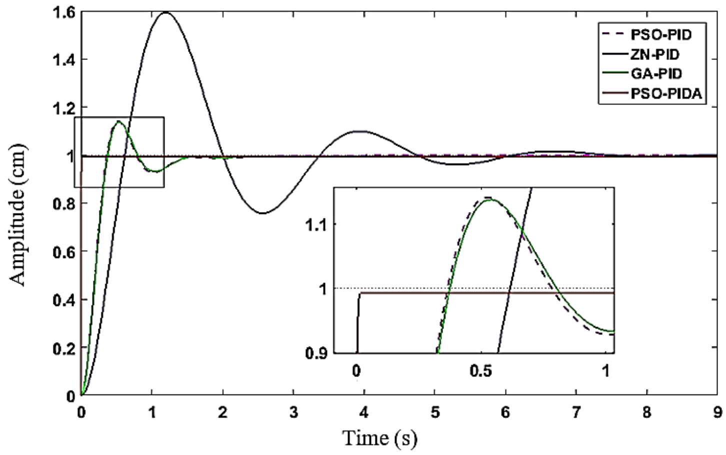
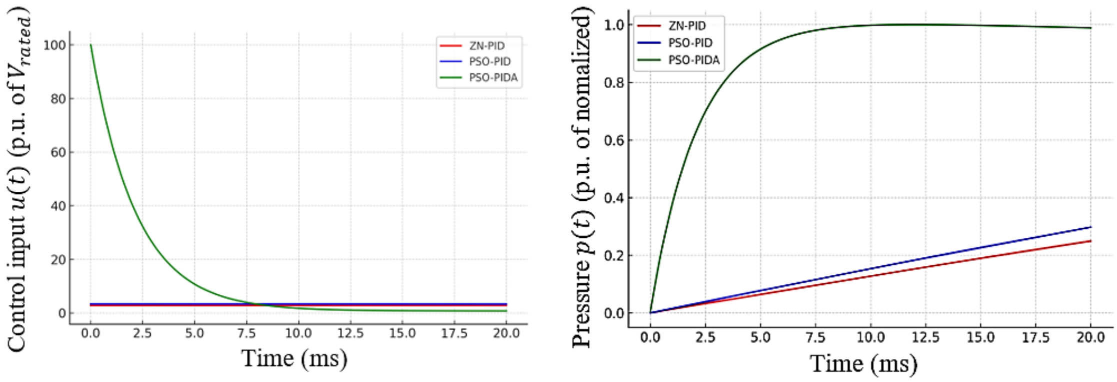


Figure 6. *System response to different controllers.*

PSO-PIDA controller makes the best control quality: Simulation shows that the system has a fast settling time of 0.0096 seconds, a rise time of 0.005 seconds, and an overshoot of 0%. The system using PSO-PIDA shortens the settling time faster than ZN-PID by 604 times and compared to GA-PID or PSO-PID by 145 times. Thus, with a third-order system, PSO-PIDA achieves high efficiency in system control quality.



a) Control input u(t) (p.u.). *b) Pressure response (p.u.).*

Figure 7. *Control input and pressure response comparison.*

Although the optimal gains of the PSO-PIDA controller reach the upper bound ($K_p = K_d = 1000$), these values are obtained within a normalized search range [0, 1000]. Physically, they correspond to a higher control sensitivity without exceeding actuator limits, because the plant model (18) already includes an internal integral term. Therefore, the controller requires a larger proportional-derivative effort to achieve rapid convergence, at the cost of higher control energy. This “trade-off” will be clarified by the control signal and pressure plots in figure 7. Accordingly, the trade-off manifests as increased control effort during the initial transient, which is acceptable for systems supplied by a constant-pressure hydraulic source.

The PSO-PIDA controller exhibits faster response with smooth pressure variation, confirming that, despite larger gain magnitudes, actuator saturation is avoided. The control input remains within ± 10% of the supply voltage limit, ensuring physical feasibility.

4. CONCLUSIONS

This paper presented a PSO-PIDA controller design for an electro-hydraulic actuation system driven by a proportional valve. The proposed approach integrates the PSO algorithm for automatic tuning of four control gains (K_p, K_i, K_d, K_a) to minimize a composite time-domain performance index. Simulation results based on the physically derived third-order hydraulic model demonstrate that the PSO-PIDA controller achieves superior dynamic characteristics compared to ZN-PID and PSO-PID controllers. Specifically, the proposed controller yields shorter settling time, smaller overshoot, and faster pressure response, while maintaining stability and acceptable control effort. The large numerical values of the PSO-PIDA parameters, as discussed, reflect the controller's higher acceleration compensation capability rather than unphysical tuning. The trade-off manifests as increased control effort during the initial transient, which is acceptable for systems supplied by a constant-pressure hydraulic source. Future work will focus on implementing the proposed controller in a real-time control platform to validate robustness under load variation and supply pressure fluctuation, and to extend the approach to multiple-axis electro-hydraulic systems.

REFERENCES

[1]. Seul Jung, Richard C. Dorf, *"Analytic PIDA Controller Design Technique for a Third Order System"*, Proceedings of the 35th Conference on Decision and Control, pp. 1–6, (1996).

[2]. Ukakimaparn, P., Pannil, P., Boonchuay, P., Trisuwannawat, T., *"PIDA Controller Designed by Kitti's Method"*, Proceedings of the 2009 ICCAS-SICE, pp. 1547–1550, (2009).

[3]. Mohamed Issa, Ahmed Abd Elbaset, Aboul Ella Hassanien, Ibrahim Ziedan, *"PID Controller Tuning Parameters Using Meta-Heuristics Algorithms: Comparative Analysis"*, Machine Learning Paradigms: Theory and Application, Springer Nature Switzerland AG, pp. 413–430, (2019).

[4]. Nguyen Trong Khuyen, Cao Tien Le, Cao Duc Sang, Vu Hai Ha, Ho Si Duong, Nguyen Duc Manh, Quach The Dung, *"Ứng dụng giải thuật di truyền cho điều khiển cân bằng hệ thống thủy lực trên xe chuyên dụng"*, Journal of Military Science and Technology, Vol. 88, pp. 42–50, (2023) (in Vietnamese).

[5]. Wei-Der Chang, Shun-Peng Shih, *"PID controller design of nonlinear systems using an improved particle swarm optimization approach"*, Communications in Nonlinear Science and Numerical Simulation, Vol. 15, No. 12, pp. 3632–3639, (2010).

[6]. Rania Hassan, Babak Cohanim, Olivier de Weck, Gerhard Venter, *"A comparison of particle swarm optimization and the genetic algorithm"*, Proceedings of the AIAA/ASME/ASCE/ASH/ASC Structures, Structural Dynamics & Materials Conference, pp. 1–13, (2005).

[7]. Kiam Heong Ang, Gregory Chong, *"PID Control System Analysis, Design, and Technology"*, IEEE Transactions on Control Systems Technology, Vol. 13, No. 4, pp. 559–576, (2005).

[8]. H. Ramezanian, S. Balochian, *"Optimal Design of a Fractional-Order PID Controller Using Particle Swarm Optimization Algorithm"*, International Journal of Control and Automation, Vol. 6, No. 1, pp. 55–67, (2013).

[9]. Wang, J., Li, X., *"Improved PSO–PID control for electro-hydraulic servo systems"*, IEEE Access, Vol. 9, pp. 13124–13136, (2021).

[10]. Rajasekharan Unnithan, A. R., Subramaniam, S. K., Gunasekaran, K., *"Improved particle swarm optimisation tuned PID position control of voice coil semi-active electrohydraulic damper"*, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, Vol. 238, No. 10, pp. 4270–4282, (2023).

[11]. Skarpetis, M. G., Kouvakas, N. D., Koumboulis, F. N., Tsoukalas, M., *"PSO-Based Robust Control of SISO Systems with Application to a Hydraulic Inverted Pendulum"*, Eng, Vol. 6, Article 146, (2025).

[12]. Guo, Y.-Q., Zha, X.-M., Shen, Y.-Y., Wang, Y.-N., Chen, G., *"Research on PID Position Control of a Hydraulic Servo System Based on Kalman Genetic Optimization"*, Actuators, Vol. 11, Article 162, (2022).

[13]. Chai Mau Shern, Rozaimi Ghazali, Chong Shin Horng, Chong Chee Soon, Hazriq Izzuan Jaafar, *"Optimization Techniques in PID Controller on a Nonlinear Electro-Hydraulic Actuator System"*, Journal of Advanced Research in Fluid Mechanics and Thermal Sciences, Vol. 56, No. 2, pp. 296–303, (2019).

[14]. Wen-sheng Xiao, Guang-xin Li, Chao Liu, Li-ping Tan, *"Improved Particle Swarm Optimization for Solving PID Control of Electro-Hydraulic Servo System"*, Biomedical Journal of Scientific & Technical Research, Vol. 51, No. 2, Article ID 008066, (2023).

TÓM TẮT

Tổng hợp bộ điều khiển PSO-PIDA thích nghi tham số với thuật toán tối ưu bầy đàn cho hệ điều khiển chân kích thủy lực

Bài báo trình bày thiết kế bộ điều khiển PSO-PIDA cho hệ chấp hành điện - thủy lực sử dụng van tỉ lệ, được cấp áp suất ổn định từ nguồn thủy lực độc lập. Mô hình toán học của hệ được xây dựng từ các nguyên lý vật lý cơ bản, bao gồm động học van, độ nén của dầu và chuyển động piston, dẫn đến hàm truyền bậc ba đặc trưng cho hệ. Thuật toán Particle Swarm Optimization (PSO) được sử dụng để xác định tối ưu các hệ số của bộ điều khiển PIDA nhằm cực tiểu hóa chỉ tiêu chất lượng tổng hợp theo miền thời gian. Kết quả mô phỏng cho thấy bộ PSO-PIDA cải thiện rõ rệt đáp ứng quá độ, giảm độ quá điều chỉnh và tăng ổn định áp suất so với các bộ điều khiển ZN-PID và PSO-PID. Các hệ số điều khiển có giá trị lớn phản ánh khả năng bù gia tốc của bộ điều khiển, giúp rút ngắn thời gian xác lập và tăng độ chính xác mặc dù biên độ điều khiển tăng. Kết quả nghiên cứu khẳng định tính khả thi và hiệu quả của bộ điều khiển PSO-PIDA, làm cơ sở cho các thử nghiệm thực nghiệm tiếp theo.

Từ khóa: Chấp hành điện - thủy lực; Tối ưu PSO; Điều khiển PIDA; Van tỉ lệ; Điều khiển phi tuyến.