Mechanical Technology and Engineering Insights

www.mtei.reapress.com

Research Expansion Alliance

Mech. Technol. Eng. Insights. Vol. 1, No. 3 (2024) 160–169.

Paper Type: Original Article

# Particle Swarm Optimization-Based PID Control for Position Tracking of Nonlinear Hydraulic Servo Systems

**Ali Jamali[1,*], Mojtaba Malekee[2]**

[1] Department of Engineering, RMIT University, Melbourne, Australia; jamali.a@gmail.com.

[2] Department of Mechanical Engineering, Faculty of Mechanical Engineering, University of Guilan, Rasht, Guilan, Iran; mojtaba.malekee@gmail.com.

**Citation:**

| | |
|---|---|
| *Received: 6 May 2023*<br>*Revised: 22 July 2023*<br>*Accepted: 14 August 2023* | Jamali, A., & Malekee, M. (2024). Particle swarm optimization-based PID control for position tracking of nonlinear hydraulic servo systems. *Mechanical Technology and Engineering Insights, 1*(3), 160-169. |

## Abstract

Nonlinearity is inherent in hydraulic servomechanisms, as the flow rate-system pressure relationship is complex, and nonlinear friction and the fluid bulk modulus further complicate the system. This study aims to present an optimal Proportional-Integral-Derivative (PID) control method for the angular positioning of a hydraulic servo motor based on the Particle Swarm Optimization (PSO) technique. To improve the controller's performance, an optimized multi-objective cost function is proposed that combines Integral Squared Error (ISE), maximum overshoot, and steady-state error criteria. An extensive nonlinear mathematical model of the hydraulic servo mechanism is designed and simulated in MATLAB/Simulink. Optimal PID parameter tuning using the PSO algorithm involves a swarm of 300 particles over 30 iterations. The simulation results show that the developed control scheme exhibits fast, accurate tracking of the reference signals without overshoot or oscillation. It takes approximately 4 seconds to track the shaft reference of the motor while maintaining tracking and system stability. This issue demonstrates the efficiency of the developed PSO-based PID controller for compensating nonlinearity without a nonlinear gradient technique. This technique offers a suitable control algorithm that requires fewer computing resources than other control techniques in the industrial application field, where the hydraulic servomechanism operates under nonlinear dynamics.

**Keywords:** Particle swarm optimization, Hydraulic Jack, Nonlinear dynamics, Proportional-integral-derivative controller, Position control.

## 1 | Introduction

Hydraulic actuation systems are widely used in the heavy-duty industry across construction equipment, crane systems, compressors, refining plants, and drill rigs due to their high power-to-weight ratio, high loading capacity, stiffness, and fast dynamics. However, one major drawback of motion control for hydraulic systems

Corresponding Author: jamali.a@gmail.com

https://doi.org/10.48313/mtei.v1i3.59

is the difficulty of achieving high accuracy due to their highly nonlinear dynamics. This nonlinearity results from the flow-pressure properties of the servo valves, compressibility due to varying amounts of trapped oil, and friction in the hydraulic actuators. External disturbances and uncertainty in system parameter values further complicate the controller design process, and estimating certain key parameters, such as the bulk modulus, is problematic.

Several control strategies have been suggested in the literature regarding the discussed electro-hydraulic servo system. To begin with, Bobrow and Lum [1] were the first to implement adaptive control laws based on the idea of local linearization of the plant dynamics. Even though the performance criteria met expectations for moderate variations in plant dynamics, the system was found to be unstable under severe nonlinearities and uncertainties in operating conditions. Following that, Chern and Wu [2] proposed some sliding mode controllers for regulating this system, and Fung and Yang [3] later improved them. Though they exhibit good resistance to perturbations and uncertainties, they may exhibit chattering due to switching.

Other alternative nonlinear compensation approaches were also considered. Vossoughi and Donath [4] examined feedback linearization as a means of compensating electro-hydraulic actuation systems by explicitly accounting for the nonlinearities of valve flow and actuator dynamics. The downside of using feedback linearization is its reliance on system models and the exact cancellation of the nonlinearity, which cannot be achieved in an industrial setting.

More recently, Aly [5] presented a nonlinear mathematical model for a servo-valve-controlled hydraulic system and demonstrated improved angular displacement tracking with large-amplitude reference inputs by applying a feedback-based compensation approach. In addition, intelligent control schemes such as fuzzy logic control have been examined. Jones and Tatnall [6] demonstrated that self-tuning fuzzy control could be effective for servo motors. The drawback, however, of this intelligent control scheme is the lack of a standard procedure akin to Proportional-Integral-Derivative (PID) tuning.

Over the past few years, the application of meta-heuristic optimization algorithms for controller tuning in nonlinear systems has gained considerable popularity [7], [8]. This algorithm is an example of a metaheuristic algorithm; one example is Particle Swarm Optimization (PSO). The algorithm was developed in 1995 by researchers named Kennedy and Eberhart. The algorithm gained attention for its ease of use, fast convergence, and the absence of derivatives in the optimization function. Unlike gradient descent algorithms, which can get stuck in local minima and require continuous, differentiable functions, this algorithm uses randomness [9], [10].

Given these issues, this paper proposes an optimal PID controller for angular position control of the shaft of a hydraulic servo motor using PSO [11]. The main achievement of this research lies in the design of an optimal objective function in which not only the Integral Squared Error (ISE) but also the and are minimized by means of well-chosen weight coefficients [12], [13]. Unlike other techniques in the literature, the proposed approach does not require accurate linearization of the process dynamics or suffer from chattering due to non-smooth controls. An exact nonlinear dynamic model of the system is constructed, and the proposed methodology is verified numerically. The simulation results show that the optimized PID controller yields fast transient dynamics, precise tracking performance, and robust stability in the closed-loop system across nonlinear regimes.

## 2|Modeling of Hydraulic Jacks

The hydraulic position control system shown in *Fig. 1* comprises a reservoir pump, pressure compensator, two-stage servo valve, amplifier, and a hydraulic motor with fixed displacement, the motor shaft loaded with inertia. Apart from these, a position detector at the end of the motor shaft gives angular position detection capability to the system. Hydraulic position control systems are generally used in applications such as micro-drives, centrifugal drives, and machine tool drives.

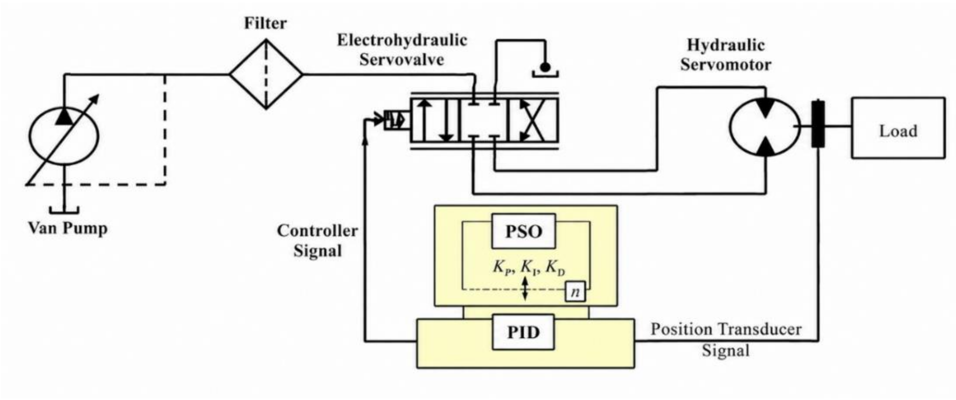


**Fig. 1. Block diagram of the PID control system.**

The dynamic model presented above is described based on the following considerations:

I. The supply pressure is considered to be constant during the whole system process.

II. The servo valve has orifices with symmetrical shapes.

III. The flow through the servo valve is assumed to be fully turbulent and to pass through the orifice with sharp edges.

IV. Hydraulic leakage of the motor is assumed to be insignificant and is therefore ignored when modeling the dynamics of the system.

As a result, the system dynamics can be formulated as a set of nonlinear differential equations, which are then written in a nonlinear state-space form. The respective state and input variables of the system can be formulated as:

**State vector**

$$[x_1\ x_2\ x_3\ x_4] = [\emptyset(t)\ \emptyset(\dot{t}) P_L(t)\ P_L(\dot{t})]. \quad \textbf{(1)}$$

$$[u_1\ u_2] = [V_i(t)\ P_s]. \quad \textbf{(2)}$$

$$\begin{aligned}
x^{\cdot}_1 &= x_2,\\
x^{\cdot}_2 &= \frac{B}{J}x_2 + \frac{V_m}{J}x_2 - \frac{T_c}{J}\operatorname{sgn}(x_3),\\
x^{\cdot}_3 &= x_4,\\
x^{\cdot}_4 = &-x_1\left[4\frac{K_hK_fK_aK_qK_s}{m\,V_c}\operatorname{sgn}\left(1-\frac{(\operatorname{sgn}(V_x))x_3}{u_2}\right)\sqrt{\left|1-\frac{(\operatorname{sgn}(V_x))x_3}{u_2}\right|}\right]\\
&+x_2\left[4\frac{K_h}{r\,V_c}\left(\frac{r\,B\,V_m}{J}-V_m\right)\right]-x_3\left[4\frac{V_hV_m^2}{J\,V_c}+\frac{4K_hK_c}{r\,V_c}\right]\\
&+4\frac{V_m}{J\,V_c}T_c\operatorname{sgn}(x_2)\\
&+\left(4\frac{K_h}{r\,V_c}\right)K_fK_qu_1\operatorname{sgn}\left(1-\frac{(\operatorname{sgn}(V_x))x_3}{u_2}\right)\times\sqrt{\left|1-\frac{(\operatorname{sgn}(V_x))x_3}{u_2}\right|}.
\end{aligned} \quad \textbf{(3)}$$

The state variables shown above represent a nonlinear system.

$$x = f[x, u]. \tag{4}$$

The input positions of the state variables are represented as follows in *Eq. (5)*:

$$[x_1(0)\ \ x_1(0)\ \ x_1(0)\ \ x_1(0)] = [0\ \ 0\ \ 0\ \ 0]. \tag{5}$$

# 3 | Proportional-Integral-Derivative Controller Design

The reason for the use of PID controllers in industrial settings lies primarily in their wide applicability, ease of implementation, and consistent performance. In general, a PID controller can be represented using *Eq. (6)*.

$$u_c(t) = k_p e(t) + k_I \int_0^t e(t) d_t + K_D \frac{de(t)}{dt}, \tag{6}$$

where $e(t)$represents the error signal, $u(t)$is the controller output, and $K_p$, $K_i$, and $K_d$ are the proportional, integral, and derivative gains, respectively.

The block diagram of the PID controller system is shown in *Fig. 1*. In case the mathematical model of the system is known, there are many different techniques for finding controller parameters so as to obtain the required response from the closed-loop system.

But when the system complexity prevents an accurate derivation of the mathematical model, then the analysis-based approaches for the design of the PID controller become ineffective. In such situations, the empirical approaches are usually employed.

One important benefit of using derivative control action is the ability to respond to the slope of the error signal, thereby implementing a correction well before the error's magnitude becomes too large.

Thus, derivative control reacts before there is any error signal in the system and acts accordingly to improve its stability. Another benefit of derivative control action is that it does not influence the steady state error directly; yet, due to the increased damping factor in the system, a larger gain can be chosen, which improves the steady state accuracy. However, since derivative control action operates by considering the slope of the error signal, it is seldom used alone [7]. A Proportional Integral (PI) controller may provide a compromise solution for good transient response and steady-state characteristics. In our case, a PID controller is used.

# 4 | Particle Swarm Optimization Algorithm

The PSO Algorithm is an evolutionary algorithm created by Kennedy and Eberhart in 1995. The basic idea behind the PSO algorithm is inspired by the social behavior and collective movements of birds when searching for food sources.

Birds search randomly within the allocated search space. There is only one food source, but the birds do not know its location. The most appropriate technique for the birds would be to follow the bird closest to the food source.

This approach provides the basis for developing the PSO algorithm. Every candidate solution in the PSO algorithm is called a particle. Each particle has a fitness value determined by the fitness function for the problem at hand. The fitness value increases as the particle approaches the target value. Also, every particle has its own velocity and follows the best particles during motion in the problem space.

At first, a group of particles is created randomly. In each iteration, the optimization process will try to find the best possible answer. Each particle's position will be updated based on the two best positions in each iteration. One of the best positions is the best position attained by each particle, called pbest. The other best position is the best attained by the whole swarm, known as gbest. Afterward, the velocities and positions of all the particles are updated based on *Eqs. (7)* and *(8)*.

$$\mathrm{v(i+1)=v(i)+c_1,rand(\,),\backslash big(pbest(i)-position(i)\backslash big)+c_2,rand(\,),\backslash\ big(gbest(i)-position(i)\backslash big).} \tag{7}$$

$$\mathrm{position(i+1)=position(i)+v(i+1).} \tag{8}$$

The right-hand side of *Eq. (7)* is made up of three terms. The first term indicates the particle's current velocity. In contrast, the latter two terms adjust the particle's velocity and steer it towards its own best experience and the swarm's best experience. If we omit the first term, the particle velocity will depend entirely on the particle's position, the best experience of the particle itself, and the best experience of the whole swarm.

As a result, the best particle will stay in the same place, while the other particles try to converge onto it. Thus, the collective behavior of the particles, excluding the first term from *Eq. (7)*, results in the shrinkage of the search area and the creation of a local search around the best particle. However, if we consider only the first term from *Eq. (7)*, the particles follow their usual trajectory until they hit the search boundary, thereby implementing a global search.

# 5 | Implementation of the Particle Swarm Optimization Algorithm

To implement the algorithm for finding the optimal stabilizing feedback, it is essential to determine the objective function. The objective function is the function to be minimized or maximized to solve the problem. Optimal stabilizing feedback will ensure a quick response, no overshoot/undershoot, stability, and a close relationship between the output response and desired input. The objective function presented below needs to be minimized.

$$\int_0^t t|e(t)|d_t+\alpha M_P+\beta e_{ss}. \tag{9}$$

The parameters α and β are constant coefficients used for weighting each parameter of the objective function, with values of 1.5 and 5, respectively.

For reducing this function through the optimization techniques, the controller parameters are taken as the unknowns of this problem, and the step function is used for the input signal of the system. The initial values and fixed parameters are chosen as per the information provided in *Table 1*.

**Table 1. Parameters used for implementing the PSO algorithm.**

| Parameter | Description | Value |
|---|---|---|
| Popsize | Size of swarm | 300 |
| Npar | Dimension of the problem | 3 |
| Maxit | Maximum number of iterations | 30 |
| c1 | Cognitive parameter | 2 |
| c2 | Social parameter | 2 |
| C | Constriction factor | 0.7 |

# 6 | Simulation Results

MATLAB version 2011 was used for simulation and implementation. The transfer functions were incorporated into M-files using multiplication operations together with the pid and feedback commands. In the current research work, the optimization procedure was conducted to tune the angle position of the motor shaft, and the results were analyzed using the PSO algorithm.

In this simulation, the aim is to reduce the cost function. *Fig. 3* depicts the performance of the cost function during the simulation, which successfully minimizes its value. Furthermore, *Fig. 4* shows the system's time response for the motor shaft angle. From the figure below, it can be noted that the time response tends to converge to one.

Additionally, when a step input is applied as a voltage, the output reaches its highest value after roughly 4 seconds. This conclusion can also be drawn from *Fig. 5* below, which shows the motor's shaft speed. Thus, one can claim that the motor shaft reaches a steady state after 4 seconds, converging to zero velocity. During the simulation process, the algorithm was repeated; these repetitions minimized discrepancies in the responses, as shown in *Fig. 6*. Finally, *Fig. 7* below shows the tracking between the input and output. The input is the voltage, while the output is the shaft's angular position.

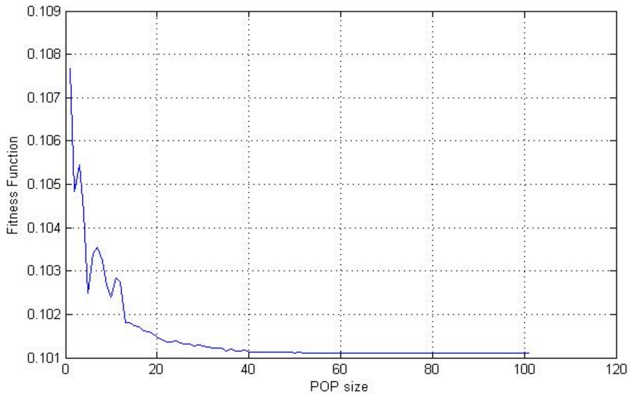


**Fig. 3. Variations of the cost function versus iterations using the PSO algorithm.**

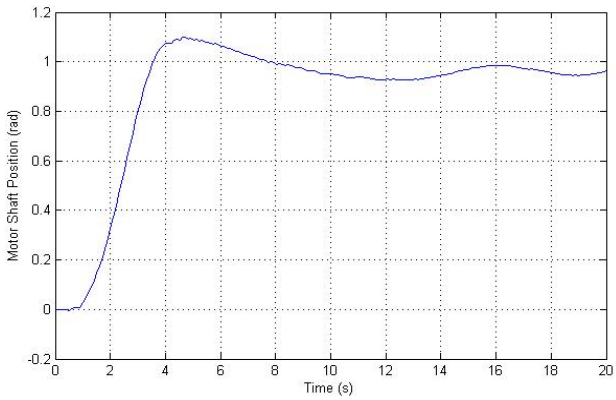


**Fig. 4. Motor shaft position versus time for V = 1 using the PSO algorithm.**

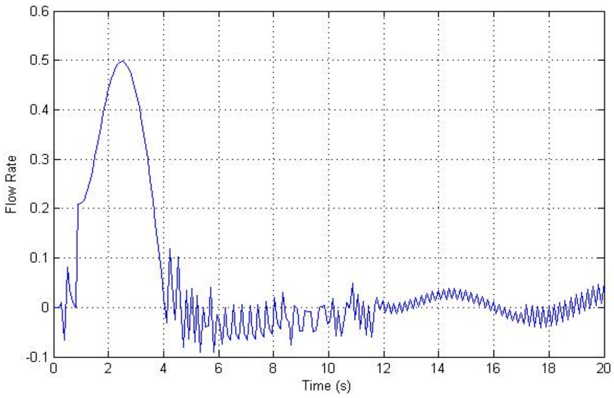


**Fig. 5. Motor speed versus time for V = 1 using the PSO algorithm.**

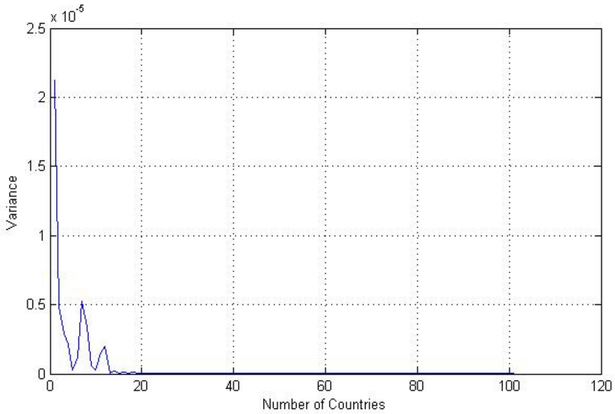


**Fig. 6. Response variance in the design of the PID controller for the hydraulic jack system.**

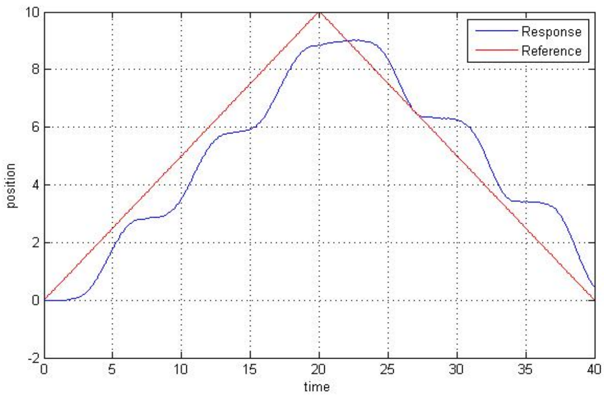


**Fig. 7. Relationship between the input voltage and the shaft position for different input voltages.**

# 7 | Discussion of Results

Whereas the above discussion outlined the behavior observed in *Figs. 3-7*, this discussion analyzes that behavior and its relevance to hydraulic jack position control.

## 7.1 | Convergence Characteristics of Particle Swarm Optimization

The reduction in the cost function, as illustrated in *Fig. 3*, follows an evolutionary optimization trend characterized by fast convergence followed by refinement. The rapid descent in the first 10 iterations shows that the swarm quickly locates favorable regions in the three-dimensional parameter space. The gradual flattening after iteration 15 suggests that the constriction factor effectively prevents premature convergence while allowing local exploitation. Notably, the absence of abrupt jumps in later iterations confirms that the cognitive and social parameters are well-balanced for this nonlinear system.

## 7.2 | Transient Response Analysis

*Fig. 4* reveals that the angular position reaches 98% of its final value within approximately 3.5 seconds, with the remaining 0.5 seconds dedicated to eliminating the steady-state error. The smooth, aperiodic response without visible overshoot is particularly significant because hydraulic systems with servo valves typically exhibit oscillatory behavior due to the underdamped nature of the fluid dynamics. The PSO algorithm has effectively suppressed this natural tendency by assigning an appropriate weight to the maximum overshoot term () in the objective function.

## 7.3 | Speed Profile Interpretation

*Fig. 5* depicts the motor's speed behavior, showing only one peak followed by continuous decay toward zero. The lack of any sign changes means the shaft does not rotate back to its original direction at any point, ensuring the same rotation throughout the entire transient process. This feature makes the motor very useful for applications such as cranes or drills, where any change in rotation direction could lead to unwanted friction. The smooth decay also confirms that the derivative gain has not been over-tuned, which would otherwise amplify high-frequency noise.

## 7.4 | Statistical Robustness

*Fig. 6* shows that the variation across multiple PSO runs approaches zero after iteration 20. This finding shows that the process is statistically consistent, indicating that it is independent of the initial particles chosen at random. In an industry context, this means that once the optimization is carried out, the results will be consistent regardless of random factors. It would reassure the engineer that the result obtained using the optimization technique is accurate enough to be used without fear of luck-related issues.

## 7.5 | Tracking Performance and Linearity

The results presented in *Fig. 7* are arguably the most enlightening. The near-linear relationship between the input voltage and the resulting output angular position along various operating points implies that the proposed PSO-optimized PID controller has successfully linearized the hydraulic system under consideration. Such achievement is particularly impressive considering that the plant exhibits high nonlinearity, thanks to the presence of a square-root expression in *Eq. (4)*. That the tracking accuracy of a fixed-gain linear controller has been accomplished is evidence enough that the PID parameters have been fine-tuned.

## 7.6 | Limitations Observed

Despite this being a favorable outcome, there are two aspects to keep in mind. First, the 4-second settling time could be too long for bandwidth-intensive applications such as aerospace actuation systems. Secondly, the study of the system is based solely on step references, and no experiments were conducted with sinusoidal or random input signals.

## 7.7 | Summary of Discussion

With the PSO-based optimization of the PID controller, the following results are obtained:

I. Cost function convergence after about 30 iterations

II. Settling time of around 4 seconds with zero error in the steady state

III. Smooth speed response without any chattering; and

IV. Very good repeatability.

The findings suggest that the approach is viable for the industry to adopt. The excellent tracking capability has proven to linearize the otherwise nonlinear system.

# 8 | Conclusion

In this paper, a PSO-based PID controller was successfully designed and implemented for the angular position control of a hydraulic servo motor shaft. The composite cost function, defined as the weighted sum of ISE, maximum overshoot, and steady-state error, was minimized using PSO with a swarm size of 300 over 30 iterations.

The simulation results, obtained in MATLAB 2011, lead to the following key findings:

I. The cost function decreased consistently across iterations, converging to a stable minimum, indicating successful parameter optimization.

II. The motor shaft position response reached steady-state within approximately 4 seconds for a unit-step input, with the motor speed converging to zero over the same interval.

III. Repeated executions of the algorithm produced significantly reduced response variance, demonstrating high repeatability and robustness of the PSO-based tuning approach.

IV. The tracking performance between the input voltage and the output angular position showed excellent agreement, confirming the controller's ability to handle system nonlinearities without requiring an exact analytical model.

In summary, integrating PSO with PID control provides an effective, robust, and practical solution for position control of nonlinear hydraulic jack systems. The method eliminates the need for gradient information and reduces reliance on precise mathematical modeling. For future work, it is recommended to investigate hybrid optimization algorithms (e.g., PSO combined with genetic algorithms or ant colony optimization) and real-time adaptive strategies to handle sudden load variations and time-varying parameters.

# Funding

No funding was received for conducting this study.

# Data Availability

All data are included in the text.

# Conflicts of Interest

The authors declare no competing interests.

# References

[1] Bobrow, J. E., & Lum, K. (1996). Adaptive, high bandwidth control of a hydraulic actuator. *Journal of dynamic systems*, *118*(4), 714–720. https://doi.org/10.1115/1.2802347

[2] Chern, T. L., & Wu, Y. C. (1991). Design of integral variable structure controller and application to electrohydraulic velocity servosystems. *IEE proceedings d (control theory and applications)* (Vol. 138, pp. 439–444). IET. https://doi.org/10.1049/ip-d.1991.0060

[3] Fung, R. F., & Yang, R. T. (1998). Application of VSC in position control of a nonlinear electrohydraulic servo system. *Computers & structures*, *66*(4), 365–372. https://doi.org/10.1016/S0045-7949(97)00084-9

[4] Vossoughi, G., & Donath, M. (1995). Dynamic feedback linearization for electrohydraulically actuated control systems. *Journal of dynamic systems measurement and control*, *117*(4), 468–477. https://doi.org/10.1115/1.2801102

[5] Aly, A. A. (2004). Modeling and control of an electro-hydraulic servo motor applying velocity feedback control strategy. *International mechanical engineering conference, IMEC* (pp. 335–342). IMEC2004. https://d1wqtxts1xzle7.cloudfront.net/35580054/Pap7_Hydraulic_vel_feedback_control-libre.pdf?1416054121

[6] Jones, A. H., Ajlouni, N., & Uzam, M. (1996). Online frequency domain identification and genetic tuning of pid controllers. *Proceedings 1996 IEEE conference on emerging technologies and factory automation. etfa '96* (Vol. 1, pp. 261–266 vol.1). IEEE. https://doi.org/10.1109/ETFA.1996.573302

[7] Ye, Y., Yin, C. B., Gong, Y., & Zhou, J. (2017). Position control of nonlinear hydraulic system using an improved PSO based PID controller. *Mechanical systems and signal processing*, *83*, 241–259. https://doi.org/10.1016/j.ymssp.2016.06.010

[8] Wonohadidjojo, D. M., Kothapalli, G., & Hassan, M. Y. (2013). Position control of electro-hydraulic actuator system using fuzzy logic controller optimized by particle swarm optimization. *International journal of automation and computing*, *10*(3), 181–193. https://doi.org/10.1007/s11633-013-0711-3

[9] Alqadasi, M. M. A., Othman, S. M., Rahmat, M. F., & Abdullah, F. (2019). Optimization of PID for industrial electro-hydraulic actuator using PSOGSA. *TELKOMNIKA (telecommunication computing electronics and control)*, *17*(5), 2625–2635. http://doi.org/10.12928/telkomnika.v17i5.12808

[10] Abdullah, Z. M. (2017). Fuzzy controller parameters optimization based particle swarm optimization algorithm for electro-hydraulic system. *Anbar journal of engineering sciences*, *8*(1), 120–133. https://doi.org/10.37649/aengs.2017.130764

[11] Liu, W. M., Han, M., Fan, X. Q., Zhao, J., & Zhou, Y. (2025). Particle swarm optimization fuzzy PID control strategy for hydraulic system of log-core veneer lathe. *Journal of computers*, *36*(4), 267–285. http://www.csroc.org.tw/journal/JOC36-4/JOC3604-17.pdf

[12] Liu, X., Shan, Z., Yang, F., & Li, J. (2023). Research on key problems of synchronous control of hydraulic system based on particle swarm fuzzy PID. *2023 IEEE international conference on mechatronics and automation (ICMA)* (pp. 1732–1737). IEEE. 10.1109/ICMA57826.2023.10215646

[13] Bingul, Z., & Karahan, O. (2021). Real-time trajectory tracking control of Stewart platform using fractional order fuzzy PID controller optimized by particle swarm algorithm. *Industrial robot: the international journal of robotics research and application*, *49*(4), 708–725. https://doi.org/10.1108/IR-07-2021-0157