Proceedings of the 12th International Conference of Control Systems, and Robotics (CDSR 2025)
July 13, 2025 - July 15, 2025 | Imperial College London Conference Center, London, United Kingdom
Paper No. 144
DOI: 10.11159/cdsr25.144

Online Tuning of PI Controller via Particle Swarm Optimization Reseeding Around Dahlin Tuning Gains for Time Delay Chemical Processes

Mehmet Uğur Soydemir[1], Savaş Şahin[1]
[1]Department of Electrical and Electronics Engineering, Faculty of Engineering and Architecture, Izmir Katip Celebi University
35620, Izmir, Turkey
mehmetugur.soydemir@ikcu.edu.tr; savas.sahin@ikcu.edu.tr

Abstract **-** This study presents an online Particle Swarm Optimization (PSO)-based approach for controlling chemical processes with delay dynamics, by adaptively tuning time-varying Proportional-Integral (PI) controller gains. In the developed method, the swarm is reseeded at each sampling instant, around the time-varying PI gains computed by a tuning mechanism, and searches for improved gains in a narrowed region. For the tuning mechanism, the Dahlin method is employed to generate the initial time-varying PI gains, which depend on the time-varying delay of the mixing tank at each control iteration. Around these gain points, new particles are sampled from a Gaussian distribution, and the most suitable gains are selected using a fitness function based on tracking error. This enables the PSO algorithm to perform fast and stable fine-tuning in a narrow parameter space, enhancing search efficiency, reducing overshoot and oscillations, while retaining the stability and proximity-to-optimality advantages of the classical tuning technique. The developed Dahlin tuning-based online PSO refined PI controller is applied to a nonlinear mixing tank system and compared with a conventional Dahlin tuning-based PI controller in simulations. The developed controller demonstrated better regulation performance than the conventional Dahlin tuning-based PI controller, and it became more robust to disturbance effects.

Keywords**:** Dahlin method, Mixing tank, Particle swarm optimization, PI controller

1. Introduction

Online tuning of PI/PID controllers aims to automatically adjust controller gains during operation, enabling adaptation to time-varying process dynamics and reducing the need for manual re-tuning. While suitable online tuning algorithms can effectively adapt controllers as system behavior evolves, many approaches require inducing oscillations or perturbations to converge to the desired parameters [1]. For instance, classical methods, such as the Ziegler–Nichols tuning technique, can drive the system into sustained oscillations, potentially causing temporary instability or process saturation [2]. In general, difficulties in online tuning include disrupting the process during tuning, the need for sufficient system stimulation for some algorithms to be successful, and maintaining robust performance under all conditions [2-6]. Therefore, online tuning methods should be supervised and used within safe operating ranges for industrial applications.

Dahlin's tuning method is a direct synthesis approach for digital controller design, providing an aperiodic response with the targeted time constant, especially in processes with first-order plus dead time (FOPDT) models. [3]. This method can perform more consistently in processes with large delays than classical PID settings [3]. Indeed, the advantages of the Dahlin method in chemical process control have been demonstrated in studies [7-8]. However, its direct use in nonlinear systems poses limitations. Applying the Dahlin method to a linearized model at each operating point can result in abrupt control actions or undesired ringing [8-9]. Hence, Dahlin method-based controllers are frequently integrated into gain scheduling or adaptive frameworks [10-11].

Analytical tuning methods, such as the Dahlin method, are typically based on a nominal process model and compute controller parameters. However, in practical applications, the actual process dynamics may deviate from the assumed model, or more optimal tuning points may exist. As a result, researchers have increasingly explored the idea of optimizing controller parameters. In this context, heuristic optimization algorithms have emerged as powerful tools for enhancing control performance beyond the limitations of model-based analytical tuning [12-15]. Particle Swarm Optimization (PSO), a heuristic approach, has been utilized in PI/PID controller tuning [16-17]. The role of PSO is to search for controller parameters that will optimize a specific performance index beyond classical tuning rules [4, 5]. In the literature, PSO-based tuning has been successfully applied to chemical processes [18-22]. Furthermore, PSO has been used to find optimal PID gains in nonlinear or unstable systems [20-21].

The advantage of PSO is that it can compensate for model errors or uncertainties in the optimization process, even if a model is required in advance. By searching within the parameters determined by analytical tuning, PSO can find a better solution based on the actual system performance metrics [6]. This approach presents a hybrid method that combines the strengths of model-based and heuristic methods: the tuning algorithm provides a fast initial controller tuning for the system, while the PSO optimizes the performance criteria by fine-tuning around this starting point [7, 8]. This paper presents a Dahlin tuning-based online PSO refined PI controller, where the initial gains are calculated using the Dahlin tuning method and subsequently refined online via a PSO algorithm. In the developed controller, PSO particles are reseeded at each iteration around externally supplied PI controller gains, computed using the Dahlin method and Gaussian sampling. By restricting the search space to a limited region around good initial values, the algorithm ensures enhanced transient performance without destabilizing the system. In contrast to conventional Particle Swarm Optimization (PSO) implementations, which disperse particles across the entire parameter space, this method capitalizes on previously identified good parameter regions. This allows for efficient tuning with a reduced number of particles and iterations.

The rest of the paper is organized as follows: Section 2 explains the Dahlin tuning method and the PSO algorithm; Section 3 describes the developed Dahlin tuning-based online PSO refined PI controller. In Section 4, the mixing tank system equations and simulation results comparing the developed controller with a conventional Dahlin tuning-based PI controller are presented. Section 5 concludes the paper.

2. Background

This section presents the background of the study.

2.1. Dahlin Tuning Method

The Dahlin tuning rule is a model-based PI/PID control tuning method explicitly developed for time-delay systems. This method operates on the system's FOPDT model and calculates the desired closed-loop behavior directly, as described in [3]. A FOPDT model is given as follows.

$$G(s) = \frac{K}{\tau s + 1} e^{-sL} \tag{1}$$

Herein, K, τ and L are process gain, time constant, and time delay, respectively. If a desired closed-loop time constant τ_c is defined, the PI controller gains and the PI controller equation might be given as follows [3].

$$K_P = \frac{\tau}{K(\tau_c + L)} \tag{2}$$

$$K_I = K_P/\tau \tag{3}$$

$$u(t) = K_P e(t) + K_I \int_0^t e(t')dt' \tag{4}$$

where K_P is proportional gain, K_I is integral gain, t is current time, $e(t)$ is instantaneous tracking error, $u(t)$ is control signal, and t' corresponds to the integral variable.

2.2. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a nature-inspired, population-based, derivative-free optimization algorithm that that iteratively evolves particles of candidate solutions by minimizing a defined fitness function [9]. Its main idea is that a a collection of particles acts together to find the best solution in the search space by mimicking the foraging behaviour of of bird flocks. Each particle represents a potential solution, and each particle updates its position (x_i) and velocity (v_i) in N-in N-dimensional search space by learning from its personal best (p_i) and the best position found by the swarm (g), gradually converging toward the global optimum. The updated velocity and position for v_i and x_i are presented by Eqs. (5) - (6).

$$v_i(t+1) = wv_i(t) + c_1r_1(p_i - x_i(t)) + c_2r_2(g - x_i(t)) \tag{5}$$
$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{6}$$

where w, c_1, c_2 are the inertia weight, cognitive coefficient, and social coefficient, respectively, and r_1, r_2 denote random numbers in [0, 1]. $wv_i(t)$ term preserves the particle's previous velocity trend, $c_1r_1(p_i - x_i(t))$ term tries to pull the particle into its best experience, and $c_2r_2(g - x_i(t))$ guides the particle towards the best experience of the swarm.

3. Developed Dahlin tuning-based online PSO refined PI controller

Considering a time delay chemical process, the developed Dahlin Tuning-based Online PSO refined PI controller closed-loop scheme is given as in Fig. 1. Herein, $r, y(t), e(t) = r - y(t)$, $K_P{}'(t)$, $K_I'(t), K_P(t), K_I(t), L(t), u(t)$ denote reference of closed-loop, output of closed-loop at time t, tracking error at time t, initial value to be used as particle center for PSO at time t, initial value to be used as particle center for PSO at time t, updated proportional gain after the PSO round at time t, updated integral gain after the PSO round at time t, a known time varying delay of chemical process at time t, and PI controller output, respectively.

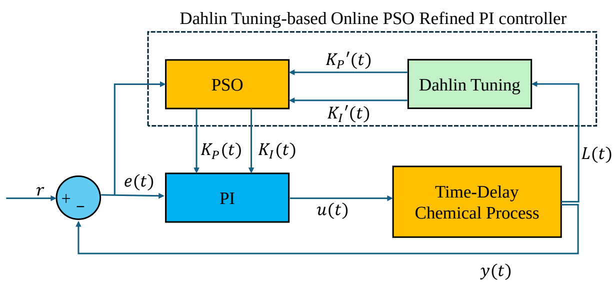

Fig. 1: The developed Dahlin tuning-based online PSO scheme.

PSO algorithm is initialized by defining an inertia weight w, cognitive coefficient c_1, social coefficent c_2, particle number N, and maximum iteration number K, initial search spreads δ_{K_P} and δ_{K_I} (standart deviations) around the Dahlin-derived gains $K_P{}'$ and the $K_I{}'$; each particle velocity v_i, and personal best position xb_i are set to 0. Its personal best fitness value xbv_i, and global best fitness value defined as gbv are set to infinity. Global best position of the entire swarm, defined as gb, is initialized to $[K_P'(t);\ K_I'(t)]$, which has first gains that are found using Dahlin tuning.

Following the FOPDT modelling of time delay chemical process, the transfer function is obtained as given in Eq. 1. Considering Eq. (2) and Eq. (3), the identified FOPDT model parameters K, τ, and time-varying delay $L(t)$ of the system are employed to compute the initial parameters of the PSO algorithm, according to Eqs. (7) – (8).

$$K_P{}'(t) = \frac{\tau}{K(\tau_c + L(t))} \tag{7}$$
$$K_I{}'(t) = K_P{}'(t)/\tau \tag{8}$$

To adapt to unmodeled dynamics and disturbances, then, each particle i is generated using a Gaussian distribution $\mathcal{N}$ to form a population of N particles by:

$$x_i = [K_P'(t);\ K_I'(t)] + \left[\delta_{K_P}\mathcal{N}(0,1);\ \delta_{K_I}\mathcal{N}(0,1)\right] \tag{9}$$

where x_i corresponds to the particle position of i. Assuming a one-step-ahead control action that is computed the current error $e(t) = e$ that comes from the closed-loop (Fig. 1), its integral $\int_0^t e(t')dt' = e_{int}$, i's particle proportional gain $K_{P_i}$and i's particle integral gain K_{I_i}, for each particle i, at current iteration k of the PSO algorithm, one-step ahead control action u_{pred_i} is computed as in Eq. (10) where T_s denotes the sampling time.

$$u_{pred_i}(kT_s) = K_{p_i}e + K_{I_i}e_{int} \tag{10}$$

Herein, $K_{P_i} = x_i(1)$ and $K_{I_i} = x_i(2)$. Assuming a one-step-ahead model instead of using a linear model or a complete model for faster evaluation with a low computational load, for each particle i, at current iteration k of the PSO algorithm, the fitness value J_i is defined as an absolute error for each particle.

$$J_i(kT_s) = \left|e - u_{pred}(kT_s)\right| \tag{11}$$

For each particle i, at current iteration k, xbv_i, xb_i, gbv and gb values are updated according to J_i. Considering Eqs. 5 and 6, the new position and new velocity are computed as in Eqs. (12) - (13).

$$v_i\big((k+1)T_s\big) = wv_i(kT_s) + c_1r_1(xb_i - x_i) + c_2r_2(gb - x_i) \tag{12}$$
$$x_i\big((k+1)T_s\big) = x_i(kT_s) + v_i\big((k+1)T_s\big) \tag{13}$$

where w, c_1, c_2 are the inertia weight, the cognitive coefficient, and the social coefficient, respectively, and r_1 and r_2 are random numbers in $(0, 1)$, and they are generated for each particle i. Upon the completion of the iterations for the PSO algorithm, the pair $(K_P(t), K_I(t))$ corresponding to the global best position gb is selected for use in the PI controller as in Eq. (4) in every control iteration of the closed-loop.

4. Results and Discussion

The developed Dahlin Tuning-based Online PSO refined PI controller scheme was applied to the mixing tank system.

4.1. Mixing Tank

Considering the mixing tank system in Fig. 2, two hot and cold streams enter the tank. $W_1(t)$ (kg/s), $W_2(t)$ (kg/s) show the mass flows of the hot and cold liquid streams, respectively. $T_1(t)$ (K) and $T_2(t)$ (K) are the hot liquid temperature and the cold liquid temperature. $T_3(t)$ (K) is the temperature inside the tank, while $T_4(t)$ (K) is the temperature value measured by the sensor at the end of a 38.1 m long pipe that exits the tank. TT, TC, and the Set Point correspond to the temperature transmitter, the temperature controller, and the reference value for the temperature transmitter. The controller output adjusts cold liquid mass flow by controlling a valve.

Certain assumptions have been made on the system, including the assumption that the liquid volume in the tank is constant, that the tank components are well mixed, that the tank and pipe are well insulated, and that the temperature transmitter is calibrated to convert temperature values between 310.93 K and 366.48 K to fractions. The equations for the system are presented below [10].

$$W_1(t)C_{p1}T_1(t) + W_2(t)C_{p2}T_2(t)\text{-}(W_1(t) + W_2(t))C_{p3}T_3(t) = V\rho C_{v3}\frac{dT_3(t)}{dt} \tag{14}$$

$$T_4(t) = T_3(t - L(t)) \text{ and } L(t) = lA\rho/(W_1(t) + W_2(t)) \tag{15}$$

$$\frac{dV_p(t)}{dt} = [m(t) - V_p(t)]/\tau_{v_p} \tag{16}$$

$$W_2(t) = \frac{500}{60}C_{VL}V_p(t)\sqrt{G_f \Delta P_v} \tag{17}$$

$$\frac{dTO(t)}{dt} = \frac{[(T_4(t) - 310.93)/55.55 - TO(t)]}{\tau_T} \tag{18}$$

Herein, C_{p1}, C_{p2}, and C_{p3} (J/(kg · K)) denote the constant-pressure specific heat capacity of the liquids, and ρ (kg/m^3) is the density of the mixing tank fluids. V (m^3) represents the liquid volume, while C_{v3} (J/(kg · K)) is the constant-volume specific heat capacity. The parameters l, A, L correspond to the pipe length (m), pipe cross-sectional area (m^2), and time delay (s), respectively. Control input $m(t) \in [0, 1]$ is the fractional valve command, τ_{v_p} represents the time constant (s) of the actuator, and $V_p \in [0, 1]$ represents the fractional valve position—zero indicating fully closed and one fully open. Valve flow coefficient is C_{VL} ((m^3/s)/Pa$^{0.5}$), G_f is the specific gravity, and ΔP_v (Pa) is the pressure drop across the valve. Finally, $TO(t) \in [0, 1]$ is the fractional temperature transmitter output, and τ_T (s) is the time constant of the temperature transmitter.

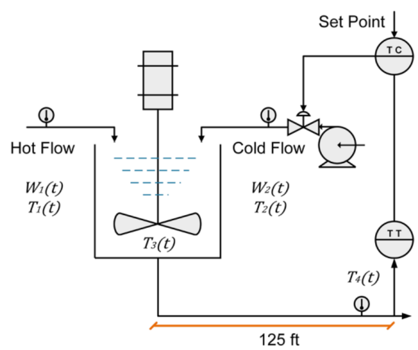

Fig. 2: Mixing tank [11].

4.2. Simulation Results

The FOPDT model of the mixing tank, as presented in subsection 4.1, is derived from the process reaction curve and is given as follows [24-25].

$$G(s) = \frac{X(s)}{U(s)} = \frac{-46.8}{139.2 + 1}e^{-178.2s} \tag{19}$$

where $X(s)$, $U(s)$ are the temperature transmitter output, controller output, and they are the deviation variables that are fractional expressions. Therefore, $u(t) = m(t) - \overline{m}$, $X(t) = TO(t) - \overline{TO}$ and tracking error $e(t) = r - TO(t)$ are taken [10]. Herein, $\overline{m}$, $\overline{TO}, r$ are the steady-state values of the controller output, temperature transmitter output, and reference of the closed-loop as in Fig. 1. Therefore, $u(t) \in [-0.478, 0.522]$ is in a controller design. The mixing tank model has been developed using Eqs. (14) - (18) in the Simulink environment. The parameters of the equations have been obtained from [10]. T_1 and T_2 are set to 393.15 K and 283.15 K. In the simulation, $W_1(t)$ has been varied over time as a disturbance effect. In the developed Dahlin Tuning-based Online PSO refined PI controller, following parameter assignments are made: for the PSO part, $c_1, c_2, N, K, \delta_{K_p}, \delta_{K_p}, T_s$ are set to $0.5, 1, 10, 20, 0.05, 0.01, 1$ s, respectively; whereas for the Dahlin tuning part, considering Eqs. (7) - (8) and (19), the time constant 139.2 s, $K = -46.8$, $\tau_c = L(t)$ s [3] and time varying delay $L(t)$ (s) is computed as in Eq. (15). The simulation time in the Simulink environment is 36000 s, the sampling time is 1 s, and r is 0.5. The developed controller is compared to a conventional Dahlin tuning-based PI controller [10] and its parameters are computed as $K_P = -0.5$ and $K_I = -0.0035$. Tracking performance of controllers for transmitter output $TO(t)$, time varying delay $L(t)$ and time evolution of $W_1(t)$ are given in Fig. 3, Fig. 4 and Fig. 5. Comparison of controllers regarding overall error, steady state error in terms of mean absolute error (MAE), settling time according to 2% criterion, and overshoot percentage is given in Table 1 [12].

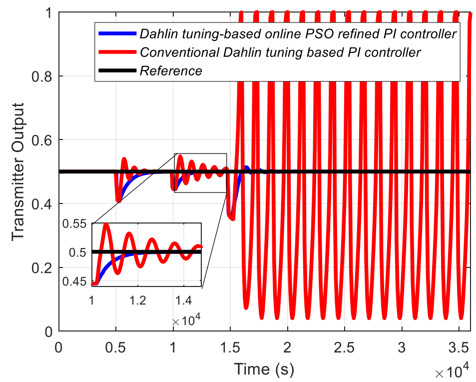

Fig. 3: Tracking performance of controllers for transmitter output $TO(t)$.

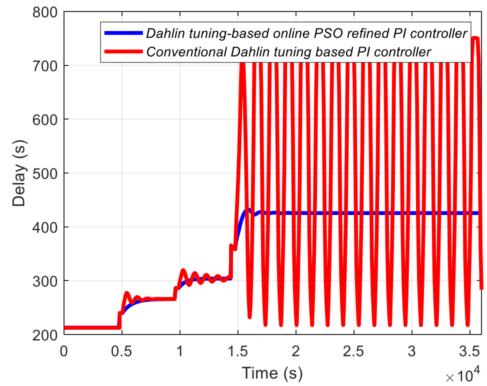

Fig. 4: Time evolution of time varying delay $L(t)$.

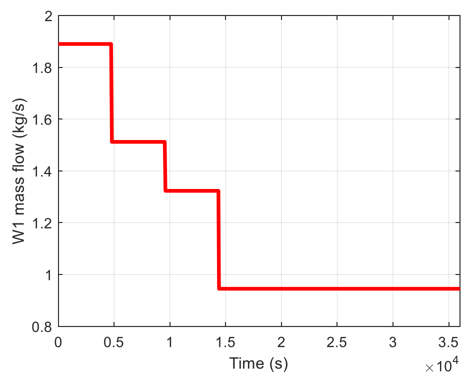

Fig. 5: Time evolution of $W_1(t)$.

While the developed Dahlin tuning-based online PSO refined PI controller demonstrates better performance in terms of overall tracking error, steady-state error, and overshoot percentage compared to the conventional Dahlin tuning-based PI controller, it does require a longer settling time to reach its final value. This trade-off suggests that Dahlin tuning-based online PSO refined PI controller sacrifices a degree of responsiveness to achieve tighter regulation and reduced oscillation. The settling time values in Table 1 represent settling times when the first disturbance effect is introduced as in Fig. 5, but subsequently developed Dahlin tuning-based online PSO refined PI controller demonstrates a smaller settling time for later disturbances compared to other controller as in Fig. 3, it can be inferred that the developed Dahlin tuning-based online PSO refined PI controller, while less effective in first disturbance rejection, ultimately offers better long-term performance.

Table 1: Comparisons of tracking performance of the developed controller and the conventional Dahlin-tuning based PI controller

Controller	Overall MAE error $e(t)$	Steady State MAE error $e_{ss}(t)$	Settling time (s)	Overshoot (%)
The Developed Dahlin Tuning-based Online PSO refined PI controller	0.0072	3.1286e-07	7020	0
Conventional Dahlin tuning-based PI Controller	0.1925	0.3291	6660	7.98

The developed controller gains K_P, K_I and controller signals are given in Figs. 6, and 7, respectively. The control signal transition for the developed controller is markedly smoother and entirely free of ringing phenomena compared to the other controller, indicating better damping and robustness during disturbance changes.

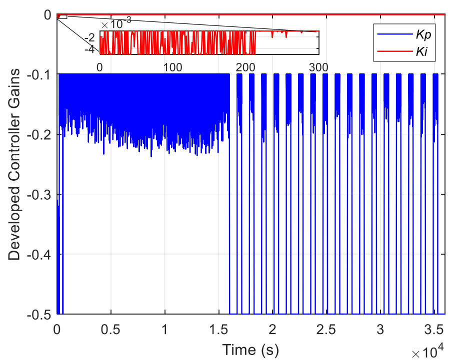

Fig. 6: Time evolution of developed controller gains $K_P(t)$ and $K_I(t)$.

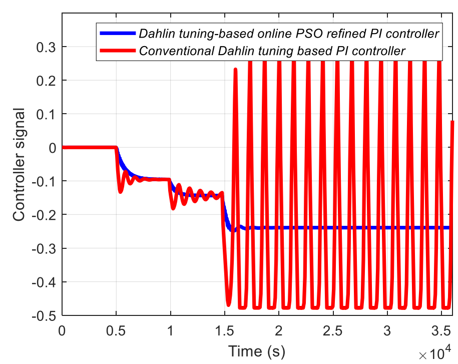

Fig. 7: Time evolutions of controller signals.

5. Conclusion

This paper presents a Dahlin tuning-based online PSO refined PI controller scheme. This scheme is a hybrid scheme that incorporates the initial gains of PSO, which are tuned using Dahlin tuning. Hence, an adjustment can be made to account for the time delay effect, and then fine-tuning can be performed using an online PSO algorithm. The developed controller is applied to a mixing tank system and compared with the conventional Dahlin tuning-based PI controller. According to simulation results, the Dahlin tuning-based online PSO refined PI controller exhibits better performance in terms of overall error, steady-state error, and overshoot, while it generally has a lower settling time compared to the other controller for disturbance changes.

References

[1] B. Roffel and B. Betlem, Process Dynamics and Control: Modeling for Control and Prediction, Wiley, 2006.

[2] K. Åström, PID Controllers: Theory, Design, and Tuning, 1995.

[3] C. A. Smith and A. B. Corripio, Principles and Practices of Automatic Process Control, Wiley, 2005.

[4] D. T. Dongshu Wang and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing*, vol. 22, p. 387–408, 2018.

[5] A. G. Gad, "Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review," *Archives of Computational Methods in Engineering* , vol. 29, p. 2531–2561, 2022.

[6] S. B. Joseph, E. G. Dada, A. Abidemi, D. O. Oyewola and B. M. Khammas, "Metaheuristic algorithms for PID controller parameters tuning: review, approaches and open problems," *Heliyon*, vol. 8, no. 5, 2022.

[7] E. M. El-Gendy, M. M. Saafan, M. S. Elksas, S. F. Saraya and F. F. Areed, "Applying hybrid genetic–PSO technique for tuning an adaptive PID controller used in a chemical process," *Soft Computing*, vol. 24, no. 5, p. 3455–3474, 2020.

[8] Y. Wang, J. Liu, R. L. X. Suo and E. Lu, "Application of PSO-BPNN-PID Controller in Nutrient Solution EC Precise Control System: Applied Research," *Sensors,* vol. 22, no. 15, p. 5515, 2022.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995.

[10] O. Camacho ve C. A. Smith, «Sliding mode control: an approach to regulate nonlinear chemical processes,» *Isa Transactions,* cilt 39, no. 2, pp. 205-218, 2000.

[11] C. Obando, R. Rojas, F. Ulloa and a. O. Camacho, "Dual-Mode Based Sliding Mode Control Approach for Nonlinear Chemical Processes," *ACS Omega* , vol. 8, no. 10, pp. 9511-9525, 2023.

[12] K. Ogatha, Modern Control Engineering, 5 ed., Pearson, 2010.

[13] T. Dlabač, S. Antić, M. Ćalasan, A. Milovanović and N. Marvučić, "Nonlinear Tank-Level Control Using Dahlin Algorithm Design and PID Control," *Applied Sciences,* vol. 13, no. 9, p. 5414, 2023.

[14] K. J. Åström and T. Hägglund, Advanced PID Control, International Society of Automation, 2006.

[15] M. Soria, D. Benítez and O. Camacho, "2DOF-PID Adaptive Gain-Scheduling Control for Nonlinear Process with Dominant Delay," in *2024 IEEE Biennial Congress of Argentina (ARGENCON)*, 2024.

[16] N. Ajlouni, A. Osyavas, F. Ajlouni, A. Almassri, F. Takaoglu and M. Takaoglu, "Enhancing PID control robustness in CSTRs: a hybrid approach to tuning under external disturbances with GA, PSO, and machine learning," *Neural Computing and Applications* , 2025.

[17] P. Deulkar and S. Hanwate, "Analysis of PSO-PID controller for CSTR temperature control," in *2020 IEEE First International Conference on Smart Technologies for Power, Energy and Control (STPEC)*, 2020.

[18] M. Shamsuzzoha, "Closed-Loop PI/PID Controller Tuning for Stable and Integrating Process with Time Delay," *Industrial & Engineering Chemistry Research,* vol. 52, no. 36, p. 12973–12992, 2013.

[19] A. Ali and S. Majhi, "Integral criteria for optimal tuning of PI/PID controllers for integrating processes," *Asian Journal of Control,* vol. 13, pp. 328-337, 2011.

[20] N.-S. Pai, S.-C. Chang and C.-T. Huang, "Tuning PI/PID controllers for integrating processes with deadtime and inverse response by simple calculations," *Journal of Process Control,* vol. 20, no. 6, pp. 726-733, 2010.

[21] I. Clitan, M. Abrudean and V. Muresan, "The Control of an Industrial Manipulator's Positioning System Using Dahlin Algorithm," *Procedia Technology,* vol. 19, pp. 541-546, 2015.

[22] W.-D. Chang and S.-P. Shih, "PID controller design of nonlinear systems using an improved particle swarm optimization approach," *Communications in Nonlinear Science and Numerical Simulation,* vol. 15, no. 11, pp. 3632-3639, 2010.

[23] D. C. Meena and A. Devanshu, "Genetic algorithm tuned PID controller for process control," in *2017 International Conference on Inventive Systems and Control (ICISC)*, 2017.

[24] M. A. Attia, A. Y. Abdelaziz and A. M. Mosaad, "Whale optimization algorithm to tune PID and PIDA controllers on AVR system," *Ain Shams Engineering Journal,* vol. 10, no. 4, pp. 755-767, 2019.

[25] H. Varol and Z. Bingul, "A new PID tuning technique using ant algorithm," in *Proceedings of the 2004 American Control Conference*, Boston, 2004.

[26] E. P. Astudillo-Ambrosi, J. F. Posso-Calderón and O. Camacho, "An Adaptive 2DOF-Nonlinear PID Controller for Nonlinear Processes with Dominant Delay," in *2024 IEEE ANDESCON*, 2024.