Black Sea Journal of Engineering and Science
doi: 10.34248/bsengineering.1888163

Open Access Journal
e-ISSN: 2619 – 8991

Research Article
Volume 9 - Issue 2: 952-961 / March 2026

# PID CONTROLLER TUNING FOR HEAT EXCHANGER SYSTEMS: A COMPARATIVE STUDY OF CLASSICAL, MODEL-BASED, OPTIMIZATION, AND REINFORCEMENT LEARNING APPROACHES

**Sertaç SAVAŞ[1]***

[1]*Erciyes University, Faculty of Engineering, Department of Mechatronics Engineering, 38039, Kayseri, Türkiye*

**Abstract:** Heat exchangers are among the fundamental components of industrial processes, and effective temperature control is critical for process efficiency and product quality. This study presents a comparative analysis of PID controller tuning methods for a heat exchanger system from four different paradigms. As the classical approach, Ziegler–Nichols (ZN); as the model-based approach, Internal Model Control (IMC); as the metaheuristic optimization approach, Particle Swarm Optimization (PSO); and as the reinforcement learning approach, Soft Actor-Critic (SAC) are investigated. For the ZN and IMC methods, a single run is executed using fixed hyperparameters, whereas a two-stage methodology is followed for the PSO and SAC methods. Hyperparameter selection is performed via random search, evaluating 20 configurations and selecting the parameters that yield the lowest ITAE. Using the chosen hyperparameters, 20 independent runs are conducted, and statistical analysis is performed. For all tuning methods, the controller's tracking performance for step, sinusoidal, triangular, and square-wave reference signals is computed using RMSE, IAE, ISE, and ITAE metrics. The results show that the PSO-PID method achieves the lowest error metrics for all reference signals. In the step response, PSO provides 90.8% improvement in ITAE and 25.2% improvement in RMSE compared to ZN. The Wilcoxon rank-sum test indicates that the differences between PSO and SAC are statistically significant for most metrics ($P<0.05$). The controller obtained via the IMC method exhibits a slow response due to the system's large time constant and substantial phase lag for periodic signals. The SAC method shows higher variance than PSO but delivers better performance than classical methods. Overall, the study reveals the strengths and weaknesses of various approaches and provides guidance on method selection for industrial heat exchanger control. The outputs also demonstrate that the PSO algorithm is an effective and reliable method for PID parameter tuning in slow, time-delay systems such as heat exchangers.

**Keywords:** PID controller tuning, Heat exchanger, Ziegler–Nichols (ZN), Internal Model Control (IMC), Particle Swarm Optimization (PSO), Soft Actor-Critic (SAC).

***Corresponding author:** Erciyes University, Faculty of Engineering, Department of Mechatronics Engineering, 38039, Kayseri, Türkiye
**E mail:** sertacsavas@erciyes.edu.tr (S. SAVAŞ)
Sertaç SAVAŞ https://orcid.org/0000-0001-8096-1140

**Received:** February 13, 2026
**Accepted:** March 09, 2026
**Published:** March 15, 2026

**Cite as:** Savaş, S. (2026). PID controller tuning for heat exchanger systems: a comparative study of classical, model-based, optimization, and reinforcement learning approaches. *Black Sea Journal of Engineering and Science*, *9*(2), 952-961.

## 1. Introduction

In industrial processes, temperature control is critical for product quality, energy efficiency, and process safety. Heat exchangers are key process equipment widely used across sectors such as the chemical and petrochemical industries, food processing, power generation, and HVAC (Heating, Ventilation, and Air Conditioning) systems. The dynamic behavior of these systems can often be modeled using second-order plus dead-time (SOPDT) transfer functions. Compared to first-order systems, SOPDT systems exhibit more complex dynamics, which makes controller design more challenging.

Proportional–Integral–Derivative (PID) controllers are preferred in the vast majority of industrial control systems due to their simple structure, ease of implementation, and wide applicability. However, the effective operation of PID controllers depends on the proper tuning of the $K_p$, $K_i$, and $K_d$ parameters. Over the years, a wide range of methods—from classical approaches to artificial intelligence–based techniques—have been developed to determine these parameters.

Among classical tuning methods, the most widely used is the ZN method proposed by Ziegler and Nichols (1942). This method uses empirical formulas based on the critical gain and period and does not require a mathematical model. However, ZN typically leads to high overshoot and oscillations. Among model-based approaches, the IMC method (Rivera et al., 1986) provides systematic controller design by directly using the process model. IMC offers flexibility via a single tuning parameter ($\lambda$) but may be sensitive to model mismatch.

In recent years, metaheuristic optimization algorithms have been successfully applied to PID tuning problems. PSO (Kennedy and Eberhart, 1995), inspired by swarm intelligence, can effectively explore the parameter space and converge toward near-global optima. Deep reinforcement learning algorithms provide a newer perspective for controller design. SAC (Haarnoja et al., 2018), within the maximum-entropy reinforcement learning framework, offers advantages in both sample efficiency and stability.

In this section, the literature on temperature control and energy efficiency in thermal processes, such as heat exchangers and reactor/HVAC systems, is systematically reviewed, spanning classical PID and robust/predictive control approaches to fractional-order, fuzzy/neuro-fuzzy, and metaheuristically tuned controllers, and finally to machine learning and deep reinforcement learning–based optimization methods.

Oravec et al. (2018) designed an integral-action Robust Model Predictive Control (MPC) scheme for shell-and-tube heat exchangers by accounting for fouling-induced uncertainties and compared it with a PID controller in simulations. Sallam et al. (2020) obtained a transfer function via Computational Fluid Dynamics based system identification for a cross-flow heat exchanger and designed a PID controller; PID tuning was performed using PSO to improve the closed-loop temperature response compared to classical tuning rules. Olana and Abose (2021) designed a PID-based controller for temperature regulation in a shell-and-tube heat exchanger. They addressed the design and tuning process in an applied manner for setpoint tracking. Al-Dhaifallah (2023) proposed a fuzzy fractional-order PID (FOPID) controller for a heat exchanger and investigated its performance under load and operating-condition variations, comparing it with conventional PID control. Jamil et al. (2022) provided a review of FOPID use in temperature control, discussing trends across different heating/temperature applications and comparing FOPID with conventional PID at a general level. Bobič et al. (2020) investigated the transient responses of a counterflow plate heat exchanger to inlet-temperature disturbances across different flow configurations and proposed a lumped-parameter one-dimensional model; they validated the model predictions with systematic experiments and infrared thermography (IR) measurements, showing that flow configuration has a pronounced effect on temperature transients.

Zou et al. (2023) reviewed machine learning (ML) methods used in heat exchanger modeling over the last eight years, classified them by exchanger type and factors such as fouling and flow regime, highlighted critical issues such as data quality and input selection, and discussed limitations and trends. Pai and Weibel (2022) developed ML-based surrogate models to predict heat transfer and pressure loss for internal flow channel cross-sections. They used them in cross-section shape optimization to reduce dependence on geometry-specific correlations. Tomar et al. (2024) established a PLC/SCADA-based implementation for heat exchanger temperature control and tuned FOPID and PID controllers using metaheuristic optimization. Ouyang et al. (2023) optimized PID parameters for Tank Reactor temperature control using an improved sparrow search algorithm, achieving more robust control under coupling and hysteresis. Sun et al. (2022) proposed an optimization approach based on online control system reconfiguration for long-term energy savings in heat exchanger networks, accounting for slow, time-varying changes due to fouling and updating the control structure and tuning online while maintaining continuous controllability. Maya-Rodriguez et al. (2023) proposed a metaheuristically tuned neuro-fuzzy (NF) controller for temperature control of a chemical reactor in biodiesel production. They compared it with a classical PID controller, targeting improved error-based performance and control energy under disturbances. Zhang and Tan (2025) proposed an entropy-based information–elimination–enhanced deep reinforcement learning method (EntropySAC) for building HVAC optimization in the Energym simulation environment. They compared it with classical control approaches and with Deep Deterministic Policy Gradient (DDPG) and SAC.

In this study, a comprehensive comparison of PID tuning methods for heat exchanger temperature control is presented using four different approaches: ZN (classical), IMC (model-based), PSO (metaheuristic optimization), and SAC (deep reinforcement learning). The methods are evaluated on step, sinusoidal, triangular, and square-wave reference signals using RMSE, IAE, ISE, and ITAE metrics. The results are statistically analyzed using the Wilcoxon rank-sum test.

## 2. Materials and Methods

### 2.1. Heat Exchanger System

The heat exchanger system considered in this study (Franklin et al., 2021) has a typical industrial structure that transfers heat between steam and water. In the system shown schematically in Figure 1, steam enters the chamber through a controllable valve at the top, transfers heat to the water flowing through the tube bundle, and exits at the bottom. Water flows through a labyrinth-like tube path inside the chamber while receiving heat from the steam.

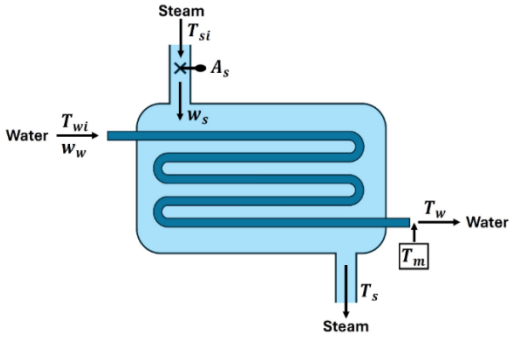


**Figure 1.** Schematic structure of the heat exchanger.

Heat transfer in the exchanger is modeled based on fundamental thermodynamic principles. Heat energy flow is proportional to the temperature difference across the medium and is expressed by equation 1.

$$q = \frac{1}{R}(T_1 - T_2) \quad (1)$$

Here, $q$ denotes the heat energy flow [J/s], $R$ is the thermal resistance [°C/W], $T$ denotes temperature, and $(T_1 - T_2)$ is the temperature difference between two points [°C]. The net heat input changes the temperature depending on thermal capacity, as given by equation 2.

$$\dot{T} = \frac{1}{C}q \quad (2)$$

where $C$ denotes thermal capacitance. The energy balance on the steam side is given by the differential equation in equation 3.

$$C_s\dot{T}_s = A_s K_s c_{vs}\,(T_{si} - T_s) - \frac{1}{R}(T_s - T_w) \quad (3)$$

In this equation, $C_s = m_s c_{vs}$ is the thermal capacitance of the steam, $A_s$ is the valve opening area, $K_s$ is the valve flow coefficient, $c_{vs}$ is the specific heat of steam, $T_{si}$ is the inlet steam temperature, $T_s$ is the outlet steam temperature, $R$ is the average thermal resistance, and $T_w$ is the outlet water temperature. Similarly, the energy balance on the water side is expressed as in equation 4.

$$C_w\dot{T}_w = w_w c_{cw}\,(T_{wi} - T_w) + \frac{1}{R}(T_s - T_w) \quad (4)$$

where $C_w$ is the thermal capacitance of water, $w_w$ is the water mass flow rate, $c_{cw}$ is the specific heat of water, and $T_{wi}$ is the inlet water temperature. Because the measurement sensor is placed at a certain distance below the outlet, there is a time delay of $t_d$ seconds between the measured temperature and the actual outlet temperature, as given in equation 5.

$$T_m(t) = T_w(t - t_d) \quad (5)$$

After linearization and Laplace transformation, the transfer function relating the steam valve opening $A_s$ to the measured outlet water temperature $T_m$ is obtained as in equation 6.

$$G(s) = \frac{T_m(s)}{A_s(s)} = \frac{Ke^{-t_d s}}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad (6)$$

The physical parameters of the heat exchanger system used in this study are given in Table 1. Table 2 gives the transfer function parameters computed from these physical parameters. Based on these values, the process gain is high ($K = 47.62$ °C), meaning that small valve-opening changes can cause significant changes in outlet temperature, potentially leading to excessive overshoot under aggressive control actions. The first time constant is $\tau_1 = 201$s; thus, the settling time is on the order of $4\tau_1 \approx 800$s, reflecting the typically slow dynamics of industrial heat exchangers. The time-constant ratio $\tau_1/\tau_2 \approx 20$ suggests the system behaves practically like a first-order process. The delay-to-first-time-constant ratio is low ($t_d/\tau_1 \approx 0.025$), which facilitates controllability; however, $t_d/\tau_2 \approx 0.5$ indicates that delay will be influential in the fast dynamics.

**Table 1.** Physical parameters of the heat exchanger

| Parameter | Symbol | Value/Unit |
|---|---|---|
| Steam thermal capacitance | $C_s$ | 2000 J/°C |
| Water thermal capacitance | $C_w$ | 21000 J/°C |
| Thermal resistance | $R$ | 0.10 °C/W |
| Water mass flow rate | $w_w$ | 0.50 kg/s |
| Specific heat of water | $c_{cw}$ | 4200 J/(kg·°C) |
| Valve flow coefficient | $K_s$ | 1.0 kg/s |
| Specific heat of steam | $c_{vs}$ | 2000 J/(kg·°C) |
| Temperature difference | $\Delta T_s$ | 50 °C |
| Time delay | $t_d$ | 5.0 s |

**Table 2.** Computed transfer function parameters

| Parameter | Symbol | Value/Unit |
|---|---|---|
| First time constant | $\tau_1$ | 201.0 s |
| Second time constant | $\tau_2$ | 9.95 s |
| Process gain | $K$ | 47.62 °C |
| Time delay | $t_d$ | 5.0 s |

### 2.2. PID Controller

The PID controller is the most widely used feedback controller type in industrial control systems. It applies three fundamental operations to the error signal $e(t) = r(t) - y(t)$, defined as the difference between the reference $r(t)$ and the measured output $y(t)$. The controller output $u(t)$ is given in equation 7.

$$u(t) = K_p\,e(t) + K_i\int e(t)\,dt + K_d\,\frac{de(t)}{dt} \quad (7)$$

where $K_p$, $K_i$, and $K_d$ denote proportional, integral, and derivative gains. The proportional term responds to instantaneous error, the integral term removes steady-state error, and the derivative term provides anticipatory action based on the rate of error change. In the Laplace domain, the PID transfer function is given by equation 8.

$$C(s) = K_p + \frac{K_i}{s} + K_d\,s \quad (8)$$

#### 2.2.1. ZN Method

The Ziegler–Nichols (ZN) method is a classical and still widely used PID tuning method (Ziegler and Nichols, 1942). In the closed-loop critical oscillation approach, integral and derivative actions are first disabled ($K_i = 0$, $K_d = 0$) and only proportional control is applied. Then, the proportional gain $K_p$ is increased gradually until sustained oscillations occur. The gain at this point is defined as the ultimate gain $K_u$, and the oscillation period is defined as the ultimate period $P_u$. PID parameters are computed from $K_u$ and $P_u$ using equation 9-11.

$$K_p = 0.60 \cdot K_u \quad (9)$$

$$T_i = \frac{P_u}{2}, K_i = \frac{K_p}{T_i} \quad (10)$$

$$T_d = \frac{P_u}{8}, K_d = K_p \cdot T_d \quad (11)$$

where $T_i$ and $T_d$ are the integral and derivative time constants. Although ZN is practical because it can be applied experimentally without requiring a model, it typically produces high overshoot and an aggressive control behavior.

**2.2.2. IMC Method**

Internal Model Control (IMC) is a model-based controller design approach that directly uses the process model. Systematized by Rivera et al. (1986), this method provides direct control over the closed-loop time constant using a single tunable parameter $\lambda$. For SOPDT processes, IMC-PID parameters are computed using equations 12-15.

$$\lambda = \max(0.25 \cdot \tau_1,\ 1.7 \cdot t_d) \quad (12)$$

$$K_p = \frac{\tau_1 + \tau_2}{K \cdot (\lambda + t_d)} \quad (13)$$

$$T_i = \tau_1 + \tau_2 \quad (14)$$

$$T_d = \frac{\tau_1 \cdot \tau_2}{\tau_1 + \tau_2} \quad (15)$$

Smaller $\lambda$ values yield faster but more aggressive responses, whereas larger $\lambda$ values yield slower but safer control. In this study, because the standard formula leads to an excessively slow response due to the large $\tau_1$, $\lambda$ is set to $\lambda = \max(0.1 \cdot \tau_1,\ 0.8 \cdot t_d) = 20.1$ s.

**2.2.3. PSO Algorithm**

Particle Swarm Optimization (PSO) is a population-based metaheuristic optimization algorithm proposed by Kennedy and Eberhart (1995), inspired by flocking birds and schooling fish. In PSO, each particle represents a candidate solution in the search space and moves by learning from its own experience and the swarm's experience. Velocity and position updates are performed as in equations 16 and 17.

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (pbest_i - x_i(t)) + c_2 \cdot r_2 \cdot (gbest - x_i(t)) \quad (16)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (17)$$

where $v_i$ is the particle velocity, $x_i$ is the particle position (PID parameters), $w$ is the inertia weight, $c_1$ is the cognitive coefficient, $c_2$ is the social coefficient, $r_1$ and $r_2$ are uniformly distributed random numbers in $[0, 1]$, $pbest_i$ is the particle's personal best position, and $gbest$ is the swarm's global best position. The objective (fitness) is defined as the ITAE value obtained with the corresponding PID parameters. To prevent excessive overshoot, a penalty is applied for overshoot values above 30%.

**2.2.4. SAC Algorithm**

Soft Actor-Critic (SAC) is a deep reinforcement learning algorithm proposed by Haarnoja et al. (2018), based on the maximum-entropy reinforcement learning framework. SAC offers advantages in sample efficiency and learning stability and is effective for continuous action spaces. The main objective of SAC is to maximize policy entropy in addition to expected reward, as given in equation 18.

$$J(\pi) = \sum_{t=0}^{T} \gamma^t \mathbb{E}[r(s_t, a_t) + \alpha \cdot H(\pi(\cdot|\, s_t))] \quad (18)$$

where $\gamma \in [0,1)$ is the discount factor, $\pi$ is the policy, $r$ is the reward function, $\alpha$ is the temperature parameter (exploration–exploitation trade-off), and $H$ is the entropy function. SAC uses two Q-functions (twin critic networks) and a policy network (actor). Target networks are updated via soft updates as in equation 19.

$$\theta_{\text{target}} = \tau \cdot \theta + (1 - \tau) \cdot \theta_{\text{target}} \quad (19)$$

where $\tau$ is the soft-update coefficient and $\theta$ denotes network parameters. In this study, SAC is adapted for PID parameter optimization. The state vector consists of the normalized cost and current PID parameters. The action specifies new PID parameter values, and the reward is defined as negative ITAE.

**2.3. Hyperparameter Tuning**

Since classical methods (ZN and IMC) are formula-based, they do not require hyperparameter tuning in the strict sense. For the ZN method, $K_u = 0.957$ and $P_u = 46.473\ s$ are determined. However, appropriate hyperparameter selection is critical for PSO and SAC. A two-stage methodology is Applied. In the random search stage, 20 hyperparameter configurations are sampled for each algorithm from the search space using a uniform distribution. Each configuration is evaluated with a single run on the step reference signal, and the configuration yielding the lowest ITAE is selected. Also SAC algorithm is trained for 450 episodes. Table 3 presents the search spaces and selected values for PSO and SAC.

**Table 3.** Hyperparameter search spaces and selected values for PSO and SAC

| Algorithm | Parameter | Search Space | Selected | Description |
|---|---|---|---|---|
| PSO | $w$ (Inertia) | [0.40, 0.90] | 0.674 | Previous speed effect |
| | $c_1$ (Cognitive) | [1.00, 2.00] | 1.797 | Individual learning |
| | $c_2$ (Social) | [1.00, 2.00] | 1.006 | Swarm learning |
| | nPop (Population) | [20, 50] | 42 | Particle number |
| SAC | lr_actor (Actor LR) | [0.0005, 0.0020] | 0.00075 | Actor learning rate |
| | lr_critic (Critic LR) | [0.0010, 0.0040] | 0.00124 | Critic learning rate |
| | $\tau$ (Soft Update) | [0.0010, 0.0100] | 0.00964 | Target network update |

In the second stage, using the selected hyperparameters, 20 independent runs are performed, and the mean PID parameters are obtained. Different random seeds are used in each run to account for the stochastic nature of

the algorithms. Table 4 lists the PID controller parameters obtained for all methods. These parameters are then used for all reference-signal tracking simulations.

**Table 4.** PID controller parameters used in the study

| Method | $K_p$ | $K_i$ | $K_d$ |
|---|---|---|---|
| ZN-PID | 0.5744 | 0.0247 | 3.3365 |
| IMC-PID | 0.1765 | 0.0008 | 1.6733 |
| PSO-PID | 0.5114 | 0.0024 | 5.2278 |
| SAC-PID | 0.5366 | 0.0113 | 5.5473 |

### 2.4. Reference Signals and Performance Criteria

Four reference signals are used for comprehensive performance evaluation: step, sinusoidal, triangular, and square wave. These signals represent different operating conditions. The step signal represents abrupt setpoint changes; the sinusoidal signal represents smooth periodic variations; the triangular wave represents linear ramp changes; and the square wave simulates abrupt periodic transitions. The simulation duration is 500 s, and the sampling period is 0.5 s. For periodic signals, the frequency is set to $f = 0.01$ Hz, amplitude to $A = 5$°C, and offset to 5°C. Performance is evaluated using four error metrics given in equations 20-23.

$$RMSE = \sqrt{\frac{1}{N}\sum e^2(t)} \tag{20}$$

$$IAE = \int |e(t)|\, dt \tag{21}$$

$$ISE = \int e^2(t)\, dt \tag{22}$$

$$ITAE = \int t \cdot |e(t)|\, dt \tag{23}$$

where $e(t) = r(t) - y(t)$ is the tracking error and $N$ is the number of samples. RMSE (Root Mean Square Error) measures the root-mean-square error; IAE (Integral Absolute Error) measures the total absolute error; ISE (Integral Square Error) penalizes larger errors more heavily; and ITAE (Integral Time-weighted Absolute Error) weights late errors more strongly, reflecting settling performance. ITAE is used as the optimization objective since it penalizes late errors and promotes faster settling.

## 3. Results

### 3.1. Control Studies

Controller performances are evaluated using four reference signals. Since ZN and IMC are deterministic, results are obtained with a single run. For PSO and SAC, 20 independent runs are performed using the selected hyperparameters. Table 5 reports the results for all reference signals and methods. For PSO-PID and SAC-PID, results are reported as (mean ± std [min, max]).

**Table 5.** Performance metrics of all methods for reference signals

| Signal | Algorithm | RMSE | IAE | ISE | ITAE |
|---|---|---|---|---|---|
| Step | ZN-PID | 0.8503 | 134.06 | 355.63 | 3930.60 |
| | IMC-PID | 0.8835 | 125.77 | 384.39 | 2532.08 |
| | PSO-PID | 0.6361 ± 0.0072 [0.6247, 0.6576] | 51.73 ± 0.64 [51.08, 53.91] | 196.30 ± 4.61 [189.08, 210.19] | 362.87 ± 14.43 [347.27, 399.75] |
| | SAC-PID | 0.6685 ± 0.0481 [0.6238, 0.7993] | 81.01 ± 13.26 [57.94, 105.13] | 218.50 ± 33.67 [188.51, 313.53] | 2076.14 ± 604.89 [916.06, 3259.33] |
| Sinusoidal | ZN-PID | 3.3160 | 1485.74 | 5494.43 | 357959.60 |
| | IMC-PID | 3.6623 | 1635.17 | 6704.32 | 389467.78 |
| | PSO-PID | 2.0457 ± 0.1051 [1.8751, 2.3287] | 887.75 ± 50.31 [806.06, 1022.34] | 2091.79 ± 219.32 [1752.06, 2705.64] | 212207.67 ± 12005.63 [192655.67, 244343.32] |
| | SAC-PID | 2.4471 ± 0.5945 [1.7327, 3.5457] | 1077.70 ± 279.52 [736.35, 1585.20] | 3156.36 ± 1569.43 [1495.06, 6282.57] | 260884.08 ± 69371.55 [176788.11, 380161.59] |
| Triangle wave | ZN-PID | 2.4964 | 1018.63 | 3118.91 | 257743.05 |
| | IMC-PID | 2.9506 | 1331.52 | 4352.86 | 333083.79 |
| | PSO-PID | 1.6425 ± 0.0829 [1.5075, 1.8601] | 792.86 ± 38.62 [729.55, 893.87] | 1352.83 ± 138.50 [1136.82, 1730.76] | 200041.79 ± 9866.90 [183981.93, 226035.93] |
| | SAC-PID | 1.9402 ± 0.4567 [1.3869, 2.8289] | 906.23 ± 198.03 [665.44, 1309.95] | 1982.55 ± 957.46 [962.32, 4001.99] | 231686.74 ± 52070.61 [168265.86, 331973.85] |
| Square-wave | ZN-PID | 5.1763 | 2217.70 | 13361.52 | 540963.04 |
| | IMC-PID | 5.1901 | 2177.29 | 13436.56 | 521602.98 |
| | PSO-PID | 4.0283 ± 0.0436 [3.9549, 4.1589] | 1052.78 ± 12.96 [1038.20, 1094.03] | 8072.78 ± 176.89 [7778.44, 8606.88] | 245160.92 ± 2998.87 [241772.48, 254317.86] |
| | SAC-PID | 4.4516 ± 0.4179 [4.0079, 5.2342] | 1415.66 ± 302.51 [1108.30, 2036.09] | 9947.32 ± 1927.52 [7988.56, 13653.3] | 337187.79 ± 77814.86 [258862.25, 504942.83] |

Overall, PSO-PID exhibits the best performance across all reference signals and metrics. In the step response, PSO achieves 90.8% improvement in ITAE compared to ZN and 85.7% compared to IMC. In RMSE, PSO provides

improvements of 22–38% over ZN and 22–44% over IMC. SAC outperforms classical methods but does not produce results as consistent as PSO. Similar trends are observed for the other reference signals. Figures 2–5 show the system responses for step, sinusoidal, triangular, and square-wave references, respectively. Each figure consists of three subplots: (a) time response and reference tracking, (b) tracking error, and (c) RMSE comparison. In subplot (a), shaded regions indicate standard deviation bands for PSO and SAC. Only the transient region is shown in the plots, whereas all metrics are computed over the full simulation horizon (500 s). Also, Figure 6 shows box plot distributions for a comparison of performance metrics.

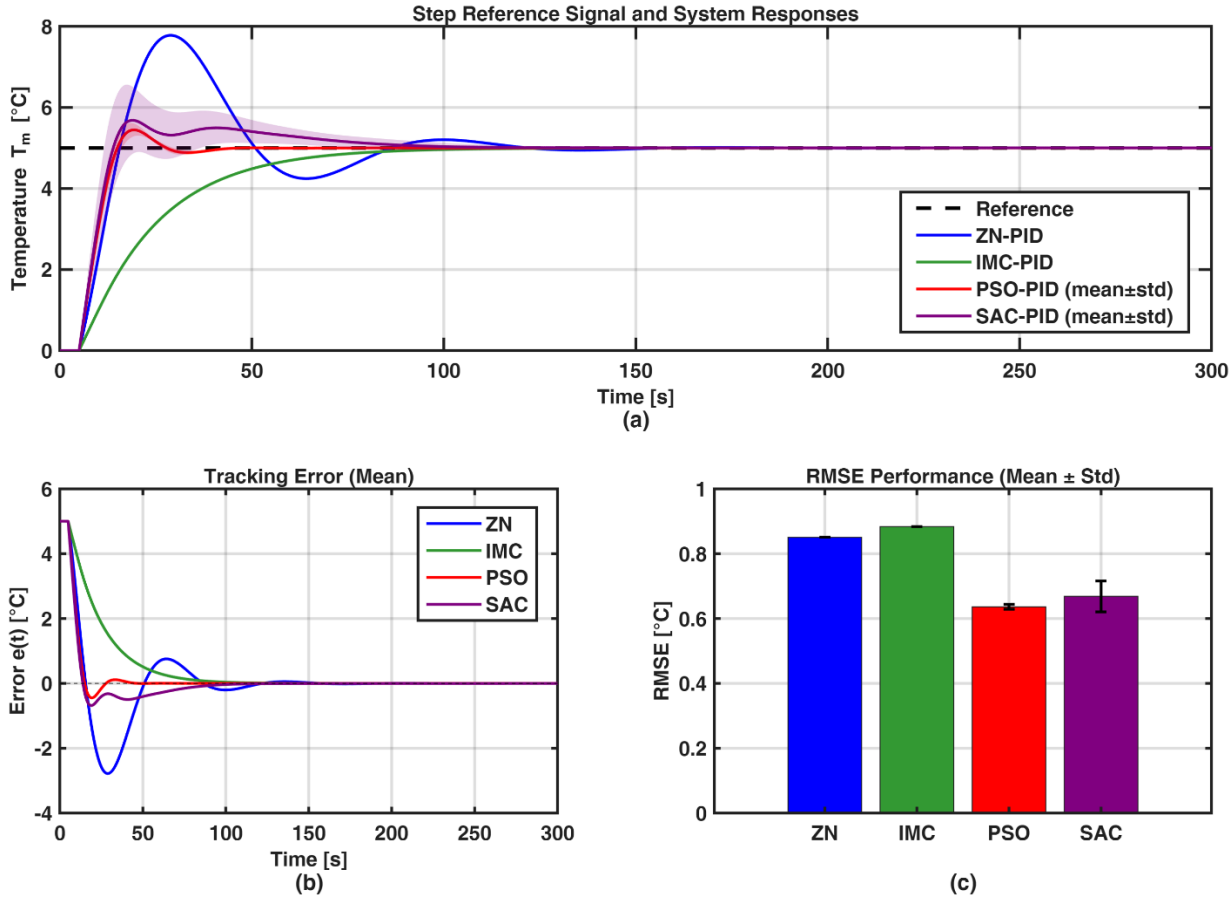


**Figure 2.** System responses for the step reference signal.

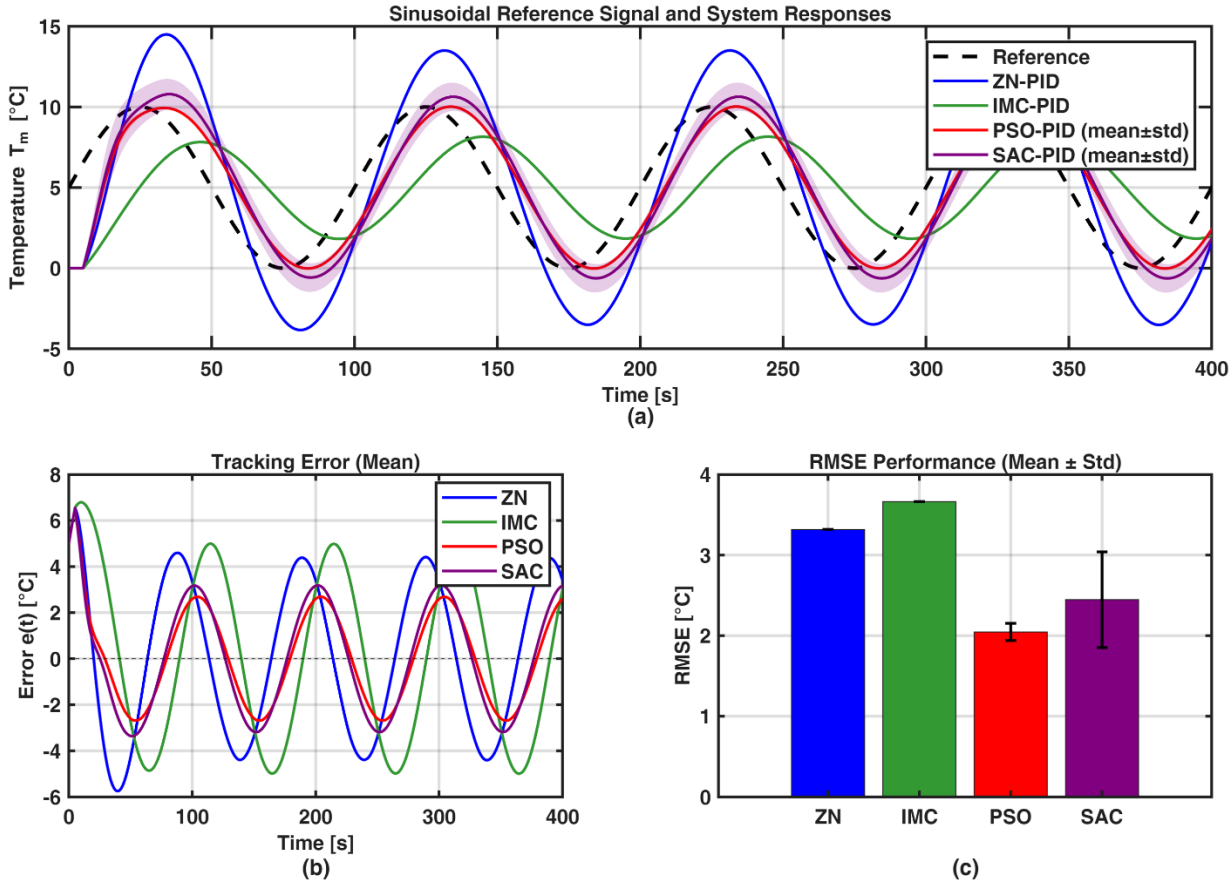


**Figure 3.** System responses for the sinusoidal reference signal.

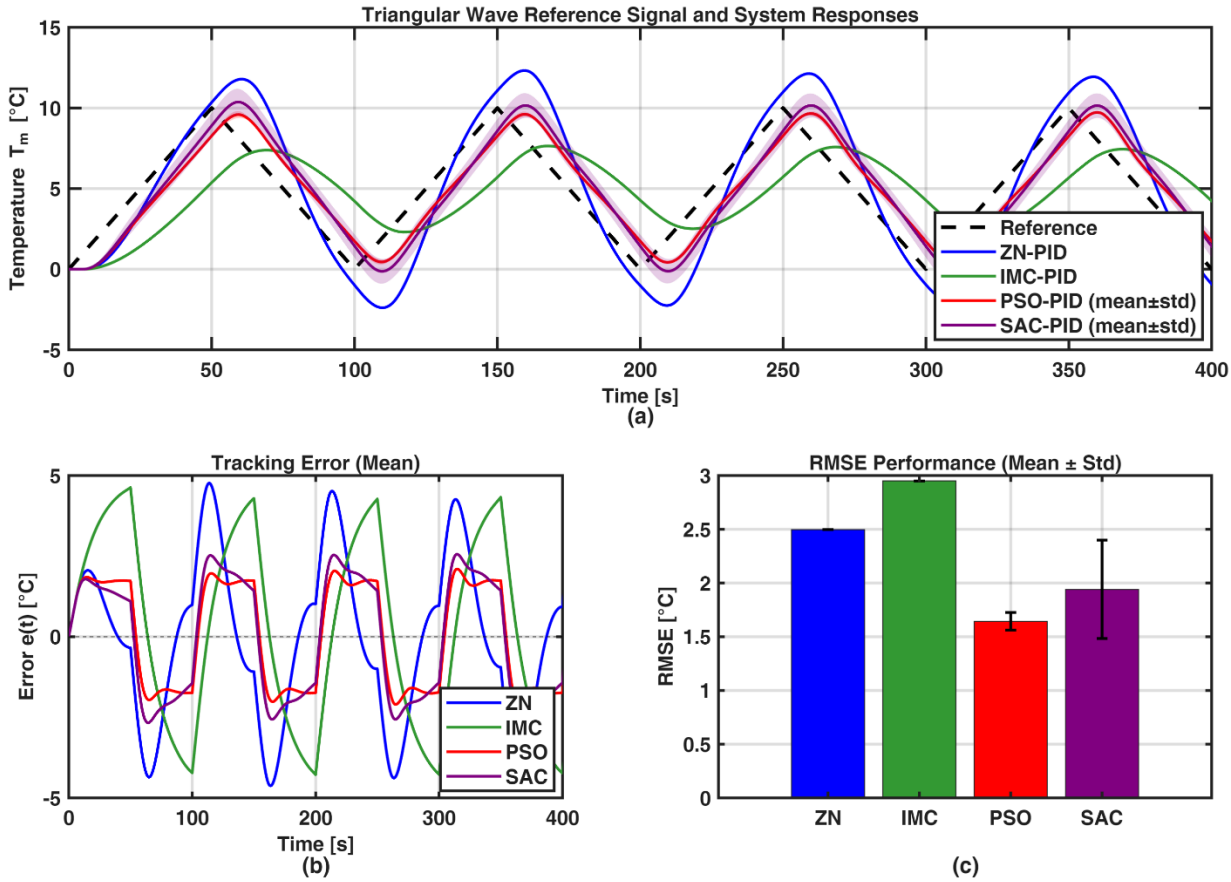


**Figure 4.** System responses for the triangular wave reference signal.

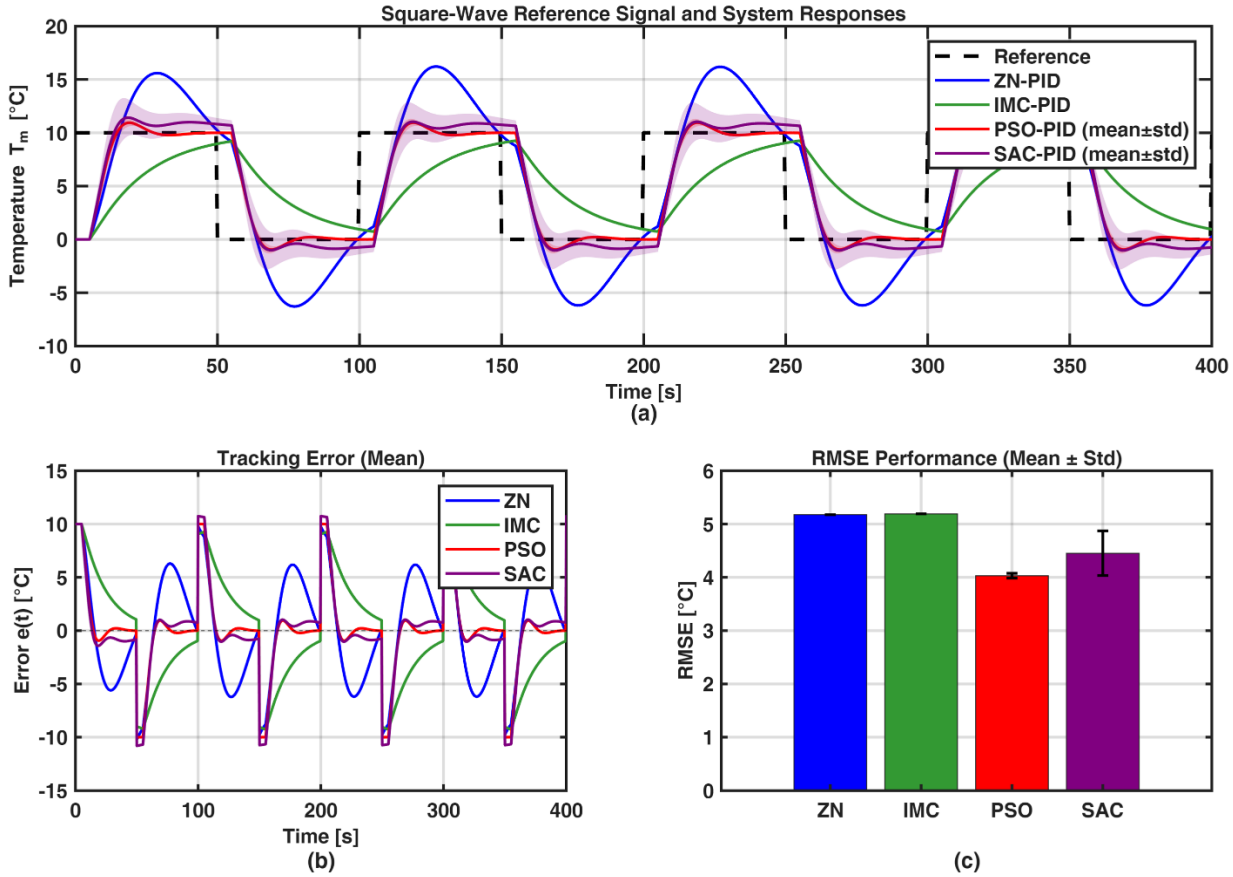


**Figure 5.** System responses for the square-wave reference signal.

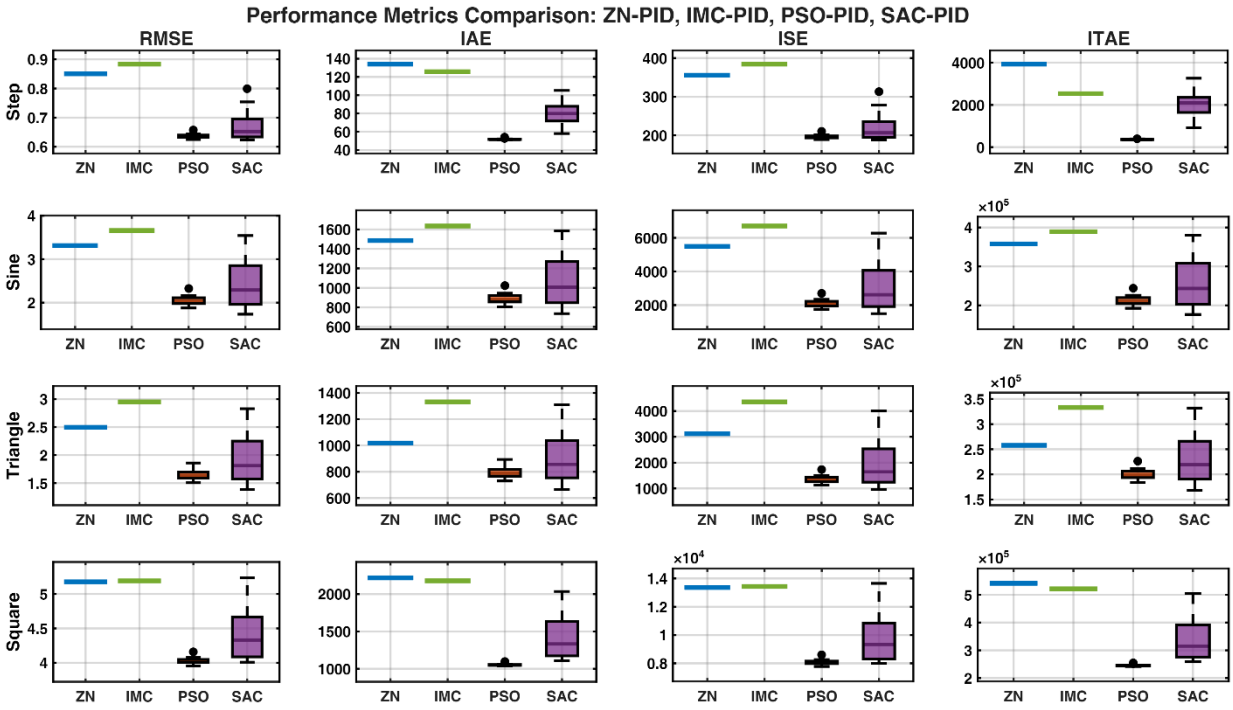


**Figure 6.** Box-plot distributions.

When we examine the figures, the step response shows that ZN-PID yields the highest overshoot and the longest-lasting oscillations. IMC-PID produces an overshoot-free but slow response and reaches steady state at approximately 100 s. PSO-PID and SAC-PID achieve fast settling with low overshoot. The narrow standard deviation band of PSO reflects high consistency, while the wider band of SAC reflects higher variance. For the sinusoidal signal, ZN-PID shows poor tracking with amplitude error and phase lag. IMC-PID exhibits a large

phase lag and trails the reference, due to reduced bandwidth caused by a large $\lambda$. PSO-PID provides the best tracking performance and follows the reference with minimal error. For the triangular wave, the sudden slope changes test derivative action. ZN-PID produces excessive overshoot at peaks, whereas IMC-PID exhibits severe phase lag. PSO-PID and SAC-PID track ramp segments well and show limited error at peak transitions. The square wave is the most challenging due to abrupt transitions. ZN-PID produces high oscillations at each transition. IMC-PID shows a very large phase lag, clearly indicating that IMC tuning is inadequate for periodic signals in this setting. PSO-PID and SAC-PID respond rapidly to transitions and provide acceptable tracking performance.

In the box plots in Figure 6, ZN and IMC values are shown as horizontal lines. The narrow box (low IQR) of PSO indicates high consistency, while the wide box of SAC indicates higher variance. Nevertheless, both PSO and SAC methods yield lower median values than the classical methods.

### 3.2. Analysis Results

Due to the stochastic nature of PSO and SAC, the statistical significance of performance differences is evaluated using the Wilcoxon rank-sum test. This nonparametric test determines whether the medians of two independent samples differ significantly. Table 6 reports the Wilcoxon test results between PSO and SAC.

The statistical analysis indicates that PSO is significantly superior to SAC for the step and square-wave signals ($P<0.05$). For the sinusoidal and triangular signals, no significant difference is found in RMSE, indicating similar tracking performance for these cases. Overall, PSO performs statistically significantly better in 16 out of 24 comparisons.

**Table 6.** Wilcoxon rank-sum test results (PSO vs SAC)

| Signal | Metric | P-value | Significant | Winner | Difference |
|---|---|---|---|---|---|
| Step | ITAE | <0.001 | Yes | PSO | 82.5% |
| | RMSE | 0.027 | Yes | PSO | 4.8% |
| Sinusoidal | ITAE | 0.041 | Yes | PSO | 18.7% |
| | RMSE | 0.053 | No | Tie | - |
| Triangular wave | ITAE | 0.096 | No | Tie | - |
| | RMSE | 0.060 | No | Tie | - |
| Square-wave | ITAE | <0.001 | Yes | PSO | 27.3% |
| | RMSE | <0.001 | Yes | PSO | 9.5% |

## 4. Discussion

This study provides a comprehensive comparison of PID controller tuning methods for a heat exchanger system from four paradigms: classical ZN, model-based IMC, metaheuristic PSO, and deep reinforcement learning–based SAC. The methods are evaluated using four error metrics on step, sinusoidal, triangular, and square-wave reference signals.

The results clearly show that PSO-PID achieves the best performance across all control scenarios. In the step response, PSO yields a dramatic performance increase, providing a 90.8% improvement in ITAE compared to ZN. This success is attributed to PSO's global optimization capability and effective exploration of the parameter space. The low standard deviation values of PSO indicate consistent and reliable outcomes. SAC-PID performs better than classical methods but produces less consistent results than PSO. The higher variance of SAC reflects the challenge of the exploration–exploitation trade-off inherent to reinforcement learning. Nevertheless, SAC offers learning capability without requiring process model knowledge or gradient calculations and thus has potential for real-time adaptive control applications.

As expected, ZN produces a response with high overshoot and oscillations, stemming from operation near the critical stability boundary and aggressive parameter selection; in practice, ZN-tuned parameters are often manually refined. IMC provides overshoot-free and safe control for the step response but exhibits significant phase lag for periodic signals.

In conclusion, this study highlights the strengths and limitations of different paradigms for PID tuning and offers guidance on method selection for industrial applications. The results indicate that metaheuristic optimization (PSO) provides an effective solution for this problem class, while reinforcement learning (SAC), though promising, requires further investigation. Future work should compare these methods on systems with different process dynamics to assess generalizability and explore hybrid approaches that combine complementary strengths to improve control performance.

**Author Contributions**

The percentages of the author' contributions are presented below. The author reviewed and approved the final version of the manuscript.

| | S.S. |
|---|---|
| C | 100 |
| D | 100 |
| S | 100 |
| DCP | 100 |
| DAI | 100 |
| L | 100 |
| W | 100 |
| CR | 100 |
| SR | 100 |
| PM | 100 |
| FA | 100 |

C= concept, D= design, S= supervision, DCP= data collection and/or processing, DAI= data analysis and/or interpretation, L= literature search, W= writing, CR= critical review, SR= submission and revision, PM= project management, FA= funding acquisition.

**Conflict of Interest**

The authors declared that there is no conflict of interest.

**Ethical Consideration**

Ethics committee approval was not required for this study because of there was no study on animals or humans.

## References


Al-Dhaifallah, M. (2023). Fuzzy fractional-order PID control for heat exchanger. *Alexandria Engineering Journal, 63*, 11–16. https://doi.org/10.1016/j.aej.2022.07.066

Bobič, M., Gjerek, B., Golobič, I., & Bajsić, I. (2020). Dynamic behaviour of a plate heat exchanger: Influence of temperature disturbances and flow configurations. *International Journal of Heat and Mass Transfer, 163*, 120439. https://doi.org/10.1016/j.ijheatmasstransfer.2020.120439

Franklin, G. F., Powell, J. D., & Emami-Naeini, A. (2021). *Feedback control of dynamic systems* (8th ed., Global ed.). Pearson.

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *arXiv:1801.01290*. https://arxiv.org/abs/1801.01290

Jamil, A. A., Tu, W. F., Ali, S. W., Terriche, Y., & Guerrero, J. M. (2022). Fractional-Order PID Controllers for Temperature Control: A Review. *Energies*, *15*(10), 3800. https://doi.org/10.3390/en15103800

Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95-International Conference on Neural Networks*. (pp. 1942-1948), 4, Perth, WA, Australia. https://doi.org/10.1109/ICNN.1995.488968

Maya-Rodriguez, M. C., Carvajal-Mariscal, I., López-Muñoz, R., Lopez-Pacheco, M. A., & Tolentino-Eslava, R. (2023). Temperature Control of a Chemical Reactor Based on Neuro-Fuzzy Tuned with a Metaheuristic Technique to Improve Biodiesel Production. *Energies*, *16*(17), 6187. https://doi.org/10.3390/en16176187

Olana, F. D., Abose, T. A. (2021). PID Temperature Controller Design for Shell and Tube Heat Exchanger. *International Journal of Engineering and Manufacturing (IJEM)*, *11*(1), 37-46. https://doi.org/10.5815/ijem.2021.01.05

Oravec, J., Bakošová, M., Trafczynski, M., Vasičkaninová, A., Mészáros, A., & Markowski, M. (2018). Robust model predictive control and PID control of shell-and-tube heat exchangers. *Energy, 159*, 1–10. https://doi.org/10.1016/j.energy.2018.06.106

Ouyang, M., Wang, Y., Wu, F., & Lin, Y. (2023). Continuous Reactor Temperature Control with Optimized PID Parameters Based on Improved Sparrow Algorithm. *Processes*, *11*(5), 1302. https://doi.org/10.3390/pr11051302

Pai, S. S., & Weibel, J. A. (2022). Machine-learning-aided design optimization of internal flow channel cross-sections. *International Journal of Heat and Mass Transfer, 195*, 123118. https://doi.org/10.1016/j.ijheatmasstransfer.2022.123118

Rivera, D. E., Morari, M., & Skogestad, S. (1986). Internal model control: PID controller design. *Industrial & Engineering Chemistry Process Design and Development, 25*(1), 252–265. https://doi.org/10.1021/i200032a041

Sallam, O. K., Azar, A. T., Guaily, A., Ammar, H. H. (2020). Tuning of PID Controller Using Particle Swarm Optimization for Cross Flow Heat Exchanger Based on CFD System Identification. In: Hassanien, A., Shaalan, K., Tolba, M. (eds) *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2019*. AISI 2019. Advances in Intelligent Systems and Computing, 1058. Springer, Cham. https://doi.org/10.1007/978-3-030-31129-2_28

Sun, L., Ren, C., & Luo, X.-L. (2022). Online control system reconfiguration towards long period energy-saving optimization of heat exchanger networks. *Journal of Cleaner Production, 367*, 132940. https://doi.org/10.1016/j.jclepro.2022.132940

Tomar, B., Kumar, N. & Sreejeth, M. (2024). PLC and SCADA based temperature control of heat exchanger system through fractional order PID controller using metaheuristic optimization techniques. *Heat Mass Transfer, 60*, 1585–1602. https://doi.org/10.1007/s00231-024-03509-5

Zhang, C., & Tan, Z. (2025). Entropy-driven deep reinforcement learning for HVAC system optimization. *J. Renewable Sustainable Energy, 17*(1), 015101. https://doi.org/10.1063/5.0238799

Ziegler, J. G., & Nichols, N. B. (1942). Optimum settings for automatic controllers. *Transactions of the ASME*, *64*(8), 759-768. http://doi.org/10.1115/1.4019264

Zou, J., Hirokawa, T., An, J., Huang, L., & Camm, J. (2023). Recent advances in the applications of machine learning methods for heat exchanger modeling—a review. *Frontiers in Energy Research, 11*, 1294531. https://doi.org/10.3389/fenrg.2023.1294531