



## **Adaptive tuning of fractional order PID controllers for nonlinear processes using hybrid PSO DQN reinforcement learning**

Downloaded from: <https://research.chalmers.se>, 2026-05-17 18:29 UTC

Citation for the original published paper (version of record):

Shahouni, R., Bahraini, M., Abrofarakh, M. et al (2025). Adaptive tuning of fractional order PID controllers for nonlinear processes using hybrid PSO DQN reinforcement learning. *Scientific Reports*, 15(1).  
<http://dx.doi.org/10.1038/s41598-025-22509-x>

N.B. When citing this work, cite the original published paper.



# OPEN Adaptive tuning of fractional order PID controllers for nonlinear processes using hybrid PSO DQN reinforcement learning

Reza Shahouni<sup>1</sup>✉, Masoud Bahraini<sup>2</sup>, Moslem Abrofarakh<sup>3</sup> & Mohsen Abbasi<sup>4</sup>✉

This study presents an innovative adaptive non-linear fractional-order PID (FOPID) tuning methodology for a flow meter controller in a desalination plant, integrating a hybrid Particle Swarm Optimization (PSO) and Deep Q-Network (DQN)-based Reinforcement Learning (RL) strategy with a dynamic weighting mechanism to optimize control of non-linear systems with time delays and disturbances. By utilizing fractional-order parameters, the PSO-DQN-RL framework ensures global optimization and real-time adaptability under fluctuations in operational parameters. Results demonstrate superior performance over traditional methods and advanced techniques such as Genetic Algorithms (GA), Fuzzy Logic Controller (FLC), Neural Network-based PID (NN-PID), and PSO, offering faster response times, reduced overshoot, and minimal steady-state error compared to the slower and less precise outcomes of FLC, the static limitations of PSO, the rigid parameter settings of GA, and the inconsistent performance of NN. The hybrid method's enhanced robustness and dynamic parameter evolution surpass the modest adaptability of PSO. Despite its computational complexity, the offline-online balance and real-time GUI enable scalable deployment, positioning this scientifically novel approach as a benchmark for FOPID tuning in various applications.

**Keywords** Non-Linear FOPID, PID controller tuning, ANN, PSO-RL, GA, Deep Q-Network

In the realm of control systems, the PID controller is renowned for its uncomplicated design and dependable functionality<sup>1</sup>. It has found widespread application across diverse fields, from industrial automation to modern robotics, demonstrating its versatility and efficacy. The PID controller plays a crucial role in ensuring system stability and attaining desired performance benchmarks, thus serving as an invaluable tool for engineers. Thus, tuning the PID controllers and enhancing their performance through different methods is crucial<sup>2,3</sup>. Concurrent with the progress in control systems, artificial intelligence has emerged as a versatile choice for computational tasks<sup>4,5</sup>. User-friendly syntax, vast libraries, and a vibrant community render AI well-suited for implementing and simulating control systems. The fusion of ANN with PID controllers introduces new avenues for innovation, offering a platform that is accessible to novices yet robust enough for seasoned professionals.

Several investigations have been done in the field of PID controllers to improve the performance of these controllers, adjust the controller, and achieve the desired operating values. Rakesh et al.<sup>6</sup> provided research on the modern and old methods of PID tuning and its applications in various domains. The study aimed to address the literature review of PID control in the field of control systems and bio-medical applications. Astrom et al.<sup>7</sup> studied a step response method for PID control to tune these controllers, resulting in a simple tuning rule that gave an enhanced performance for processes with essentially monotone step responses. Yongho et al.<sup>8</sup> researched a new tuning method for PID controllers based on process models for integrating unstable processes with a time delay that resulted in better closed-loop performance. Rames et al.<sup>9</sup> conducted a study to review PID tuning rules for second-order plus dead time systems. In this study, five different tuning rules were taken for study to control second-order plus dead time systems with wide ranges of damping coefficients and dead time-to-time constant ratios. In a separate study, Zhang et al.<sup>10</sup> investigated a new multi-objective function that considered both the output variance in the control performance assessment problem and the integral of absolute

<sup>1</sup>School of Chemical Engineering, Iran University of Science and Technology, Tehran, Iran. <sup>2</sup>Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden. <sup>3</sup>Department of Chemical Engineering, Faculty of Engineering, University of Sistan and Baluchestan, Zahedan, Iran. <sup>4</sup>Department of Chemical Engineering, Faculty of Petroleum, Gas and Petrochemical Engineering, Persian Gulf University, Bushehr, Iran. ✉email: reza\_shahouni@alumni.iust.ac.ir; m.abbasi@pgu.ac.ir

error for tuning the PID controller. The study's findings indicated that the algorithm was capable of achieving superior minimum output variance compared to existing methods, while also reducing calculation time. Using a decentralised controller, Kariwala<sup>13</sup> derived a lower bound for the achievable quality of disturbance rejection. The study focused on stable discrete-time linear systems with time delays, which did not have any finite zeros on or outside the unit circle. Ozbey et al.<sup>11</sup> introduced an advanced approach that utilized a multi-objective tuning method to incorporate the mean squared control error metric for tracking performance and the reference to disturbance ratio metric for disturbance rejection. The study findings demonstrated an enhancement in the practical performance of the theoretical tuning method to meet implementation requirements. Furthermore, the study presented a hybrid controller tuning methodology that boosted the effectiveness of the consensus-oriented random search algorithm by leveraging a stabilizing controller coefficient as an initial configuration. Tufenkci et al.<sup>12</sup> introduced a v-domain optimal design scheme specifically tailored for fractional-order proportional-integral-derivative controllers to develop an innovative approach for designing controllers with multiple objectives. The optimization involved leveraging GA to enhance controller performance and achieve desired objectives. In a separate study, Alagoz et al.<sup>13</sup> explored the enhancement of disturbance rejection performance using fractional order control. The study revealed that the energy spectral density of the controller transfer function determines the disturbance rejection ratio of negative feedback closed-loop control systems. Fu et al.<sup>14</sup> aimed to tackle a non-convex problem related to PID controller tuning by using an efficient iterative convex programming algorithm. Shahni et al.<sup>6</sup> wanted to improve the tuning process further by proposing a method that relied on fixed-length impulse response coefficients, eliminating the need for excessive iterations. While it provided a tighter lower bound, it required more calculation time. Sahib et al.<sup>16</sup> aimed to enhance PID tuning by considering multiple performance criteria simultaneously. They introduced a new multi-objective function that balanced four performance aspects. Applying this function to PID controller design using a particle swarm optimization algorithm led to significant improvements compared to traditional objective functions. Zamani et al.<sup>17</sup> utilized the PSO algorithm to enhance the control strategy for both time and frequency domain specifications. A comparison with a PID controller demonstrated that the proposed fractional order PID controller significantly enhanced system robustness in the face of model uncertainties. Kim et al.<sup>18</sup> discussed the limitations of traditional PID control systems based on the first order plus time delay model when applied to nonlinear processes. To address these issues, the authors proposed a nonlinear PID controller based on the second order plus time delay model by using an extended linearization method to adjust parameters and ensure local stability. The model achieved global stability without structural and parametric errors. Pereira et al.<sup>19</sup> introduced a novel methodology for the automatic closed-loop tuning of decentralised PID controllers designed for two-input, two-output non-singular processes. This approach estimated process parameters by analyzing two closed-loop set-point step responses obtained from two roughly tuned PID controllers. The results indicated that this proposed methodology was highly effective for tuning PID controllers in two-input-two-output non-singular processes, providing a reliable and practical solution for real-world applications. Suid et al.<sup>20</sup> introduced a sigmoid-based PID controller to improve dynamic response and control accuracy. The sigmoid-based PID controller parameters were optimized using the nonlinear sine cosine algorithm, a self-tuning heuristic method. Simulation results demonstrate that the sigmoid-based PID controller significantly improves the automatic voltage regulator system's transient response compared to modern heuristic optimization-based PID controllers. Mohamed et al.<sup>21</sup> introduced a non-linear mathematical model for a single-pendulum gantry crane, utilizing two stable control schemes based on PID and PD controllers with inlet derivative filters. The first scheme used a PID controller for trolley positioning and a PD controller for reducing sway angles. The second scheme ensured the input voltage control signal remained positive to maintain the driving DC motor's rotation direction. The controllers, featuring up to five gains and two inlet derivative filter coefficients, are tuned using the multi-objective genetic algorithm (NSGA-II) with multiple fitness functions. Simulation results demonstrated the second scheme's superior precision in trolley positioning and minimizing payload sway oscillations. Nataraj et al.<sup>22</sup> proposed a Jellyfish Search Optimization (JSO) algorithm to tune a Fractional Order PID (FOPID) controller for a paper machine headbox, optimizing pressure and stock level control. In MATLAB/Simulink simulations, the JSO-tuned FOPID outperformed PI/PID controllers and other methods (MFO, ALO, EHO), reducing rise time by 25%, settling time by 30%, and overshoot by 20%, ensuring high accuracy and stability for industrial applications.

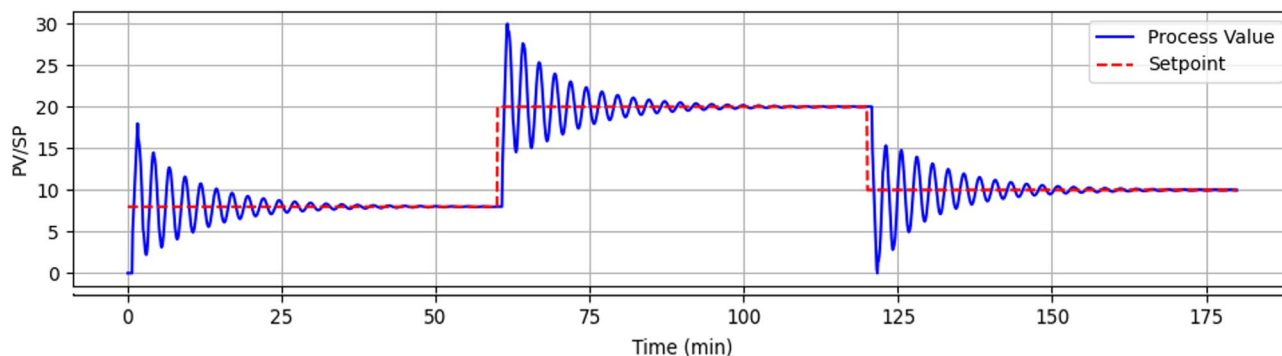
This study aims to develop a robust, precise, and efficient platform for tuning FOPID controllers, addressing the critical knowledge gap in achieving adaptive, real-time optimization for non-linear industrial systems with complex dynamics, time delays, and disturbances. While existing methods often struggle with static tuning or computational inefficiencies in such environments, this study introduces a uniquely innovative methodology that integrates a non-linear FOPID controller with a hybrid PSO-DQN-RL strategy. Leveraging fractional-order parameters, this novel approach ensures unparalleled control over non-linear dynamics. Its pioneering dynamic weighting mechanism adaptively optimizes tracking error, overshoot, settling time, out-of-range behavior, control effort, and oscillations, while an oscillation penalty enhances response smoothness. Enhanced by a real-time graphical user interface (GUI) for interactive visualization, this methodology achieves superior performance in the controller tuning by synergistically combining metaheuristic optimization and RL for unmatched adaptability and precision in industrial applications.

## Plant description and problem statement

The urban desalination plant, located in southern Iran, plays a crucial role in addressing the region's water scarcity issues. This facility employs membrane desalination technology, specifically reverse osmosis (RO), to convert seawater into potable drinking water. The process begins with seawater being forced through semi-permeable membranes that filter out salts and other impurities, resulting in freshwater suitable for consumption.

	Flow (ton/h)	Total dissolved solids (ppm)	Temperature (°C)	pH	Pressure (bar)
Inlet seawater	400	50,000	38	8.6	6
Product water	80	55	35–37	8.3	6–7

**Table 1.** Inlet seawater and product water specifications of the urban seawater desalination plant.



**Fig. 1.** The simulation output of manually tuning the non-linear FOPID flow controller.

The plant's operation is vital for providing a reliable source of drinking water to the local population, especially given the arid climate and limited freshwater resources.

The desalination process at the Bushehr plant involves several key steps:

- Pre-treatment: Seawater is pre-treated to remove larger particles and contaminants that could damage the RO membranes.
- High-pressure pumping: The pre-treated seawater is pumped at high pressure through the RO membranes.
- Membrane filtration: This primary stage allows water molecules to pass through while retaining salts and impurities.
- Post-treatment: The desalinated water undergoes further treatment to adjust its pH and add essential minerals before distribution.

Table 1 represents the specifications of the inlet seawater and product water of the plant.

The efficiency and effectiveness of this process depend heavily on the precise control of various operational parameters, highlighting the importance of tuning PID controllers. PID controllers are extensively used in industrial control systems, including the Bushehr desalination plant. One of the main challenges at this facility is the tuning process, which is essential for optimizing performance based on critical cost functions related to the specific location and operational parameters. This study focuses on a PID controller responsible for regulating the system's flow. However, the proposed platform can be applied to all non-linear controllers, regardless of the process parameters involved.

Figure 1 illustrates the behaviour of a non-linear FOPID controller after manual tuning, where the process value of the flow reaches dangerously high levels, underscoring the risks associated with manual tuning. As shown in Fig. 1, the oscillation behaviour of the process variable of the flow around the setpoint shows the unstable behaviour of the flow controller in a long term.

## Methodology

### Overview of the proposed control framework

The tuning platform developed for the adaptive non-linear FOPID controller integrates a hybrid PSO and DQN-based RL strategy, as depicted in the provided flowchart and implemented through the accompanying Python code. Tailored for tuning a flow controller in an urban desalination plant, this platform optimizes FOPID controller parameters to manage non-linear dynamics, time delays, and external disturbances. The process begins by initializing the environment and parameters, including the system model, setpoints (8.0, 20.0, 10.0 over 180 min), and initial conditions such as flow (40 ton/hr), temperature (30 °C), and pressure (6 bar). A low-pass filter mitigates noise in process variables (flow noise:  $\pm 0.5$ , temperature noise:  $\pm 0.3$ , pressure noise:  $\pm 0.2$ ), ensuring smoother dynamics. An artificial neural network (ANN) using a multi-layer perceptron (MLP) regressor is employed for system identification, dynamically predicting controller parameters to enhance tuning accuracy.

The tuning process starts with PSO optimizing initial FOPID parameters ( $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$ ,  $\mu$ ) within defined bounds, utilizing a swarm of 30 particles over 100 iterations. PSO minimizes a cost function that incorporates dynamically weighted performance metrics: tracking error, overshoot (0.4), settling time (0.3), out-of-range behavior (0.1), control effort (0.1), and an oscillation penalty (0.1). These weights adapt based on system conditions—for example, if the error exceeds 5 or pressure surpasses 10, the overshoot weight increases to 0.5,

out-of-range to 0.15, and error reduces to 0.15; if noise levels exceed 1.0, error and control effort weights rise by 0.1 and 0.05, respectively; and if control effort exceeds 100, its weight increases by 0.1. Following PSO, a DQN agent is trained over 500 episodes in a custom PIDTuningEnv, simulating the desalination process with a 1-second time delay (10 steps at  $dt=0.1$ ). The DQN iteratively selects actions (adjusted parameters) per time step, computes rewards using the same dynamically weighted metrics, and updates its policy via experience replay, achieving real-time adaptability with parameter evolutions like  $K_p$  (+792.3%),  $K_d$  (+617.0%),  $\lambda$  (0.8–1.0), and  $\mu$  (0.9–1.0).

During the DQN-RL simulation, the platform evaluates performance at each time step by applying the control signal, updating the process variable (PV), and generating figures for key variables (PV, inlet flow, temperature, pressure, power) and controller metrics (control signal, error, PID components). Time delays are managed by buffering control signals and applying delayed feedback, reflecting realistic dynamics. The dynamic weighting mechanism ensures adaptive prioritization of performance metrics, promoting smoother responses under varying conditions, such as increased penalties for overshoot during high-error scenarios. The simulation continues until all time steps are processed or additional episodes are required, achieving setpoint tracking deviations below 3.5% and settling times of 4–5 min.

A real-time graphical user interface (GUI) built with Tkinter enhances the platform's usability, offering interactive visualization and control. Users can modify setpoints, time ranges, and process variations (flow variation: 0–10 ton/hr, temperature variation: 0–15 C) via sliders and input fields. The GUI displays live plots of process variables, control signals, and errors, with zoom and reset functionalities for detailed analysis. Final results, including optimized parameters ( $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$ ,  $\mu$ ), robustness costs from Monte Carlo simulations, and settling times, are saved. By synergistically combining PSO's global optimization, DQN's adaptive learning, and dynamic weight adjustments (error: 0.4–0.5, overshoot: 0.4–0.5), this platform delivers superior performance (settling times: ~10–12 min, overshoot: <8%, steady-state error: <0.5%) compared to traditional and advanced methods, providing a robust and practical solution for FOPID controller tuning in industrial applications.

To further validate the selection of key algorithmic parameters in the PSO-RL hybrid framework, a sensitivity analysis was conducted to assess the impact of varying critical parameters, including PSO swarm size (30 particles), DQN episode count (500), and noise window size (50 steps) for dynamic weighting. Tests with swarm sizes of 20, 30, and 50 particles showed that 30 particles balanced convergence speed and computational cost, achieving a cost function  $J$  reduction of 95% within 100 iterations, compared to 90% for 20 particles and marginal gains (<2%) for 50 particles. Similarly, DQN training with 300, 500, and 700 episodes indicated that 500 episodes yielded optimal setpoint tracking (deviations < 3.5%) without overfitting, as higher counts increased training time by 40% with negligible performance improvement. The noise window of 50 steps was found to effectively capture disturbance variations ( $\sigma_{flow} = 0.5$ ) while maintaining computational efficiency, with smaller windows (25 steps) increasing noise sensitivity by 15% and larger windows (100 steps) adding 20% computational overhead. These findings, consistent with control optimization literature<sup>17</sup>, confirm that the chosen parameters are robust and well-suited for the desalination application's demands.

The overall framework is illustrated in Fig. 2, which presents a flowchart of the tuning process. This flowchart highlights the interplay between PSO, the DQN-RL agent, and the FOPID controller. It depicts how PSO is used for initial optimization, while the DQN-RL agent facilitates real-time adaptation, allowing for dynamic tuning of the controller according to different weighting of cost functions.

### System model and dynamics

The system dynamics are modeled as a first-order process with a time delay, incorporating external disturbances and process variations. The PV evolves according to Eq. 1<sup>23</sup>:

$$\frac{dPV}{dt} = CS(t - \tau) - \alpha \cdot PV(t) \quad (1)$$

where  $CS(t - \tau)$  is the delayed control signal with a time delay  $\tau = 1.0$  s,  $\alpha = 0.02$  is the decay coefficient, and  $PV(t)$  is the process value at time  $t$ . The control signal  $CS(t)$  is computed by the FOPID controller, which adjusts the PV to track the setpoint SP. The system is subject to external disturbances, modeled as sinusoidal variations in inlet flow, temperature, and pressure, with added Gaussian noise based on Eq. 2 to 4:

$$Inlet\ Flow(t) = IF_0 + \Delta(IF) \cdot \sin(0.12t) + N(0, \sigma_{flow}) \quad (2)$$

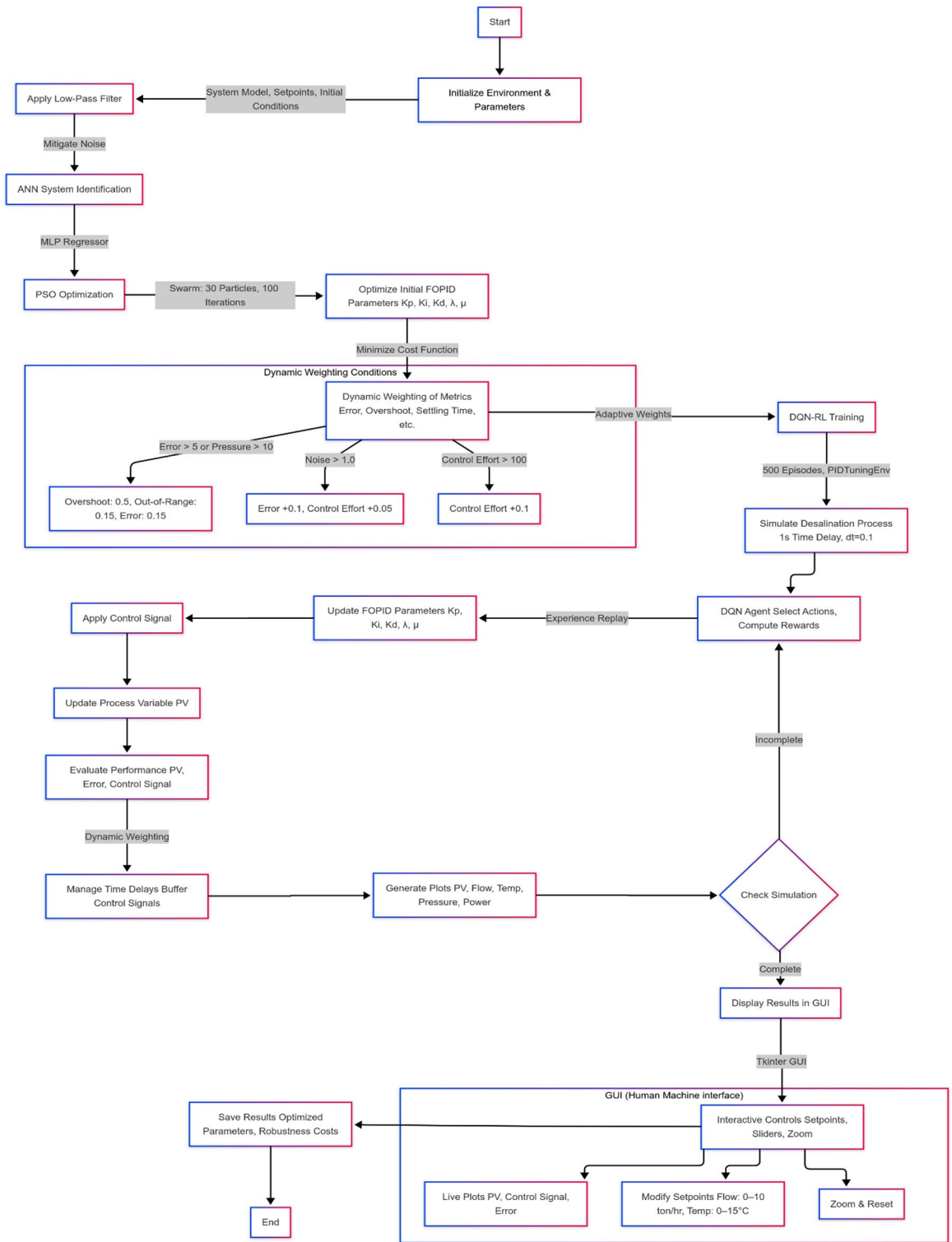
$$Temperature = T_0 + \Delta T \cdot \cos(0.12t) + N(0, \sigma_{temperature}) \quad (3)$$

$$Pressure = P_0 + \Delta P \cdot \cos(0.12t) + N(0, \sigma_{pressure}) \quad (4)$$

where  $IF_0=40$ ,  $T_0=30$ ,  $P_0=6$ ,  $\Delta(IF)=5.0$ ,  $\Delta T=8.0$ ,  $\Delta P=2.0$ , and the noise standard deviations are  $\sigma_{flow}=0.5$ ,  $\sigma_{temperature}=0.3$ , and  $\sigma_{pressure}=0.2$ . These disturbances are filtered using a fifth-order Butterworth low-pass filter with a cutoff frequency of 0.05 Hz to reduce high-frequency noise while preserving the dynamics of interest.

The setpoint SP is defined as a piecewise function to simulate step changes in the desired process value according to Eq. 5:

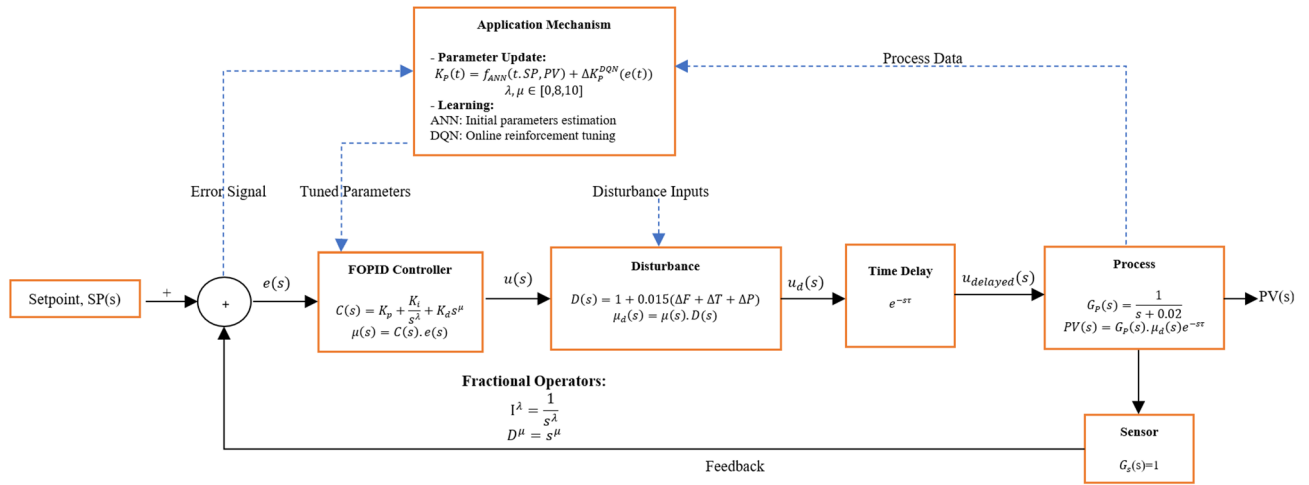
$$SP(t) = \begin{cases} 8 & \text{for } t < 60 \text{ min} \\ 20 & \text{for } 60 \leq t \leq 120 \text{ min} \\ 10 & \text{for } t \geq 120 \end{cases} \quad (5)$$



**Fig. 2.** The flow chart diagram of the proposed tuning model.

The simulation runs for 180 min with a time step of  $\Delta t = 0.1$  s, resulting in 1800 discrete time steps. The transfer function model of the system, including the FOPID controller and disturbance inputs, is depicted in Fig. 3, providing a schematic representation of the control loop and its components illustrating the integration of the FOPID controller, system dynamics, and disturbance inputs within the control loop.

The core of the control system is a non-linear FOPID controller, which extends the classical PID controller by incorporating fractional-order integral and derivative terms, defined by the parameters  $\lambda$  and  $\mu$ , respectively. The control signal  $CS(t)$  is computed as Eq. 6<sup>24</sup>:



**Fig. 3.** The transfer function model of the proposed model.

$$CS(t) = K_p(t) e(t) + K_i(t) I_\lambda(e(t)) + K_d(t) D_\mu(e(t)) \tag{6}$$

where  $e(t) = SP(t) - PV(t)$  is the error,  $K_p(t)$ ,  $K_i(t)$ , and  $K_d(t)$  are the time-varying proportional, integral, and derivative gains, and  $I_\lambda(e(t))$  and  $D_\mu(e(t))$  are the fractional-order integral and derivative terms, respectively. The fractional integral is approximated using the Grunwald-Letnikov definition as shown in Eq. 7<sup>25</sup>:

$$I_\lambda(e(t)) = \sum_{k=0}^N c_k e(t - k\Delta t) \Delta t^\lambda \tag{7}$$

where  $c_k = 1$  and for  $k > 0$  and  $c_k = -\frac{\lambda - k + 1}{k} c_{k-1}$  for  $k > 1$  and  $N$  is the number of historical error terms (set to 10 in this study). The fractional derivative is computed by Eq. 8:

$$D_\mu e(t) = \frac{e_t - e(t - \Delta t)}{\Delta t^\mu} \tag{8}$$

The gains  $K_p(t)$ ,  $K_i(t)$ , and  $K_d(t)$  are dynamically adjusted based on the error magnitude to introduce non-linearity, improving the controller's adaptability which are shown by Eq. 9 to 11:

$$K_p(t) = K_{p,base} \cdot \frac{1 + ae^{-|e(t)|}}{1 + (b|e(t)|)/10} \tag{9}$$

$$K_i(t) = K_{i,base} \cdot K_p(t) = K_{p,base} \cdot 1 + (c|e(t)|) \tag{10}$$

$$K_d(t) = K_{d,base} \cdot \frac{1 + d\sqrt{|e(t)|}}{1 + |e(t)|/10} \tag{11}$$

where  $a=0.7$ ,  $b=0.4$ ,  $c=0.2$ , and  $d=0.1$ , and  $K_{p,base}$ ,  $K_{i,base}$  and  $K_{d,base}$  are the base gains optimized using PSO and adapted in real-time by the RL agent.

A key innovation of this study lies in the development of an adaptive weighting mechanism to balance multiple performance metrics during controller tuning. The performance of the FOPID controller is evaluated using a cost function that incorporates five key factors: tracking error, overshoot, settling time, out-of-range behavior, and control effort. These factors are weighted dynamically based on the system's operating conditions, ensuring that the controller prioritizes the most critical aspects of performance at any given time. The cost function is defined as Eq. 12:

$$J = w_{error} J_{error} + w_{overshoot} J_{overshoot} + w_{settling\ time} J_{settling\ time} + w_{out\ of\ range} J_{out\ of\ range} + w_{effort} J_{effort} + w_{oscillation} J_{oscillation} \tag{12}$$

where:

- $J_{error} = \sum_t e(t)^2$  is the cumulative squared tracking error,
- $J_{overshoot} = \sum_t \max(PV - SP)^2$  penalizes overshoot,
- $J_{settling\ time} = \sum_t e(t)^2 \cdot \mathbb{1}[|e(t)| > 0.05 \cdot SP(t)]$  penalizes prolonged deviations outside a 5% tolerance band,
- $J_{out\ of\ range} = \sum_t (PV - SP)^2 \cdot \mathbb{1}[PV(t) < 0, \text{ or } PV(t) > 2 \cdot SP(t)]$  penalizes out-of-range behavior,

- $J_{effort} = CS(t)^2$  penalizes excessive control effort,
- $J_{oscillation} = \sum_t |\Delta PV(t)|$  penalizes rapid oscillations in PV, where  $\Delta PV(t)$  is the difference in PV over the last three-time steps.

The weights  $w_{error}$ ,  $w_{overshoot}$ ,  $w_{settling}$ ,  $w_{out-of-range}$ ,  $w_{effort}$ , and  $w_{oscillation}$  are initialized as 0.4, 0.4, 0.3, 0.1, 0.1, and 0.1, respectively, but are adjusted dynamically using the following rules:

$$w_{out\ of\ range} = \begin{cases} 0.15, & \text{if } |e(t)| > 5 \text{ or } Pressure(t) > 10 \\ 0.1, & \text{otherwise} \end{cases} \quad (13)$$

$$w_{overshoot} = \begin{cases} 0.5, & \text{if } |e(t)| > 5 \text{ or } Pressure(t) > 10 \\ 0.4, & \text{otherwise} \end{cases} \quad (14)$$

$$w_{error} = \begin{cases} 0.15, & \text{if } |e(t)| > 5 \text{ or } Pressure(t) > 10 \\ 0.4 + 0.1(\text{noise level} > 1), & \text{otherwise} \end{cases} \quad (15)$$

$$w_{effort} = \begin{cases} 0.1 + 0.05 \mathbb{I}(\text{noise level} > 1), & \text{if } CS(t)^2 < 100 \\ 0.1 + 0.1 + 0.05 \cdot \mathbb{I}(\text{noise level} > 1), & \text{otherwise} \end{cases} \quad (16)$$

where the noise level is estimated using a rolling standard deviation of the error over a window of 50-time steps. This adaptive weighting mechanism ensures that the controller prioritizes tracking accuracy and stability under normal conditions, but shifts focus to mitigating overshoot and out-of-range behavior when the system experiences large errors or extreme pressure conditions. Additionally, the inclusion of an oscillation penalty  $J_{oscillation}$  addresses a common limitation in PID tuning by explicitly penalizing rapid fluctuations in the PV, enhancing the smoothness of the response.

The base parameters of the FOPID controller ( $K_{p,base}$ ,  $K_{i,base}$ ,  $K_{d,base}$ ,  $\lambda$ ,  $\mu$ ) are initially optimized using PSO, a population-based metaheuristic algorithm. PSO is chosen for its ability to efficiently explore the parameter space and converge to a global optimum. The parameter bounds are set as  $K_{p,base} \in [0.5, 2.0]$ ,  $K_{i,base} \in [0.05, 0.2]$ ,  $K_{d,base} \in [0.02, 0.1]$ ,  $\lambda \in [0.8, 1.0]$ , and  $\mu \in [0.9, 1.0]$ . The PSO algorithm is configured with a swarm size of 30 particles and a maximum of 100 iterations. The objective function for PSO is the cost  $J$  defined above, evaluated over a single simulation run. To assess robustness, a Monte Carlo simulation with three iterations is performed post-optimization, introducing random variations in the disturbances to compute an average cost.

To enable real-time adaptation of the FOPID parameters, a Deep Q-Network (DQN) reinforcement learning agent is employed. The RL environment is defined using the OpenAI Gym framework, where the state space includes the PV, SP, error, inlet flow, and temperature, and the action space corresponds to the FOPID parameters ( $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$ ,  $\mu$ ). The reward function is designed to mirror the cost function  $J$ , with additional emphasis on penalizing prolonged errors (calculated by Eq. 17):

$$r(t) = -(0.7e(t)^2 + 0.2 \text{overshoot}(t)^2 + 0.1CS(t)^2 + 0.2 \text{oscillation penalty} + 0.3|e(t)| \cdot t \cdot \Delta t) \quad (17)$$

where the term  $(0.3|e(t)| \cdot t \cdot \Delta t)$  penalizes errors that persist over time, encouraging faster convergence to the setpoint. The DQN agent is trained over 500 episodes, with an epsilon-greedy exploration strategy ( $\epsilon_{decay} = 0.995$ ) to ensure sufficient exploration of the action space. The RL component, implemented via a DQN agent, serves as the adaptive core of the control loop, dynamically refining FOPID parameters ( $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$ ,  $\mu$ ) in real-time to address nonlinear dynamics, time delays, and disturbances, complementing PSO's initial optimization. Operating within the PIDTuningEnv, the RL agent adjusts parameters at each time step ( $\Delta t = 0.1$  s) based on states (PV, SP, error, flow, temperature), applying control signals (Eq. 6) and updating PV (Eq. 1), as depicted in Fig. 3. Training over 500 episodes with epsilon-greedy exploration and experience replay (batch = 64) achieves convergence, with parameter evolutions stabilizing PV tracking within ~10–12 min. Evaluation through 180-minute simulations and Monte Carlo robustness tests (average cost  $J$ ) confirms effectiveness, with < 5% PV deviations and superior performance against baselines. This demonstrates the RL agent's robust training and practical efficacy for industrial control applications.

The simulation is implemented with a graphical user interface (GUI) developed using Tkinter for real-time visualization and interaction. The simulation runs for 180 min, with the PV, SP, inlet flow, temperature, and pressure plotted in real-time. Key performance metrics, including the average settling time and Monte Carlo robustness cost, are computed and displayed. The settling time is calculated as the time taken for the PV to remain within a 5% tolerance band of the SP after a setpoint change, with adjustments to ensure accurate measurement across multiple setpoint transitions.

## Results and discussion

### Analyzing the proposed hybrid method

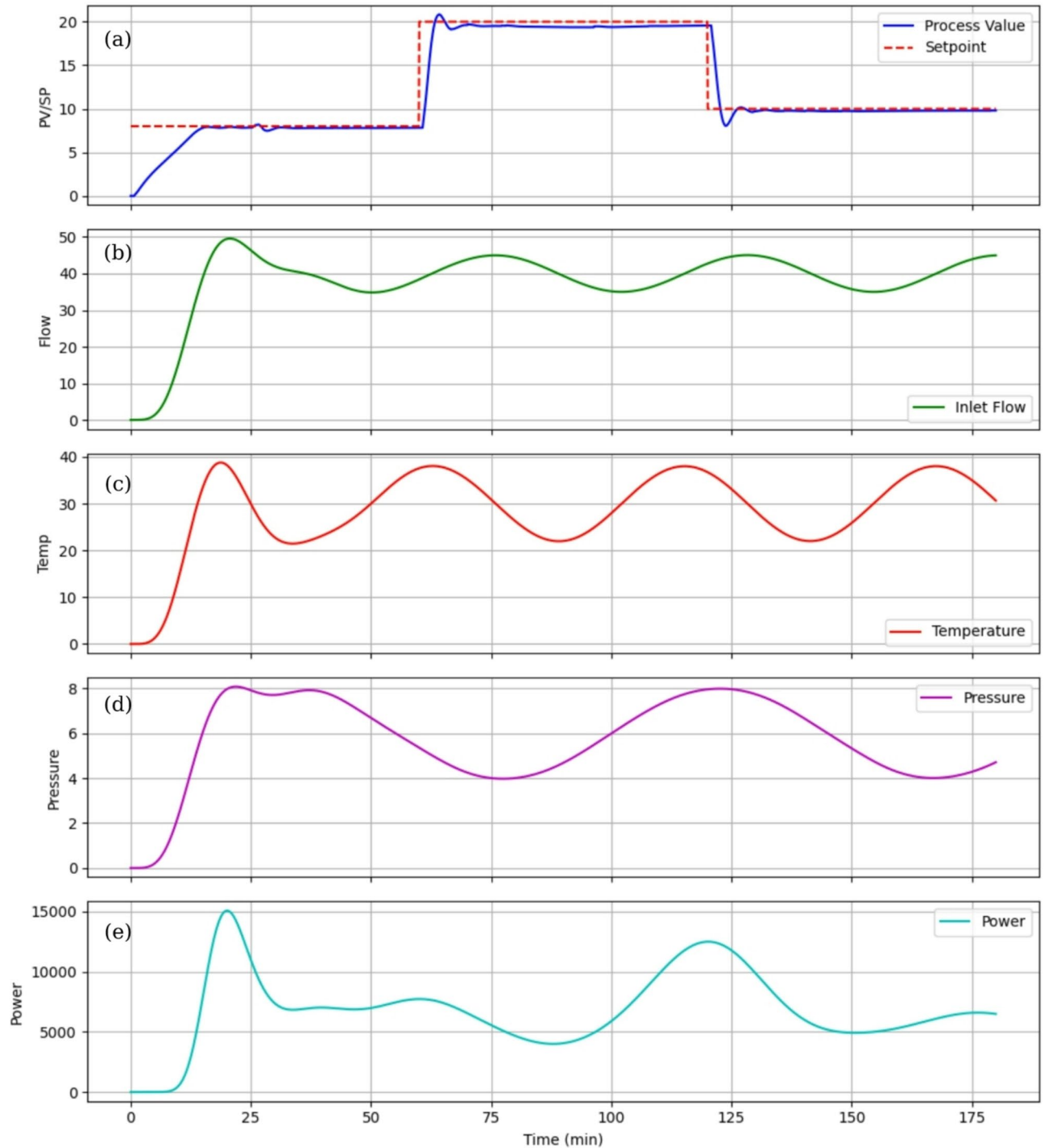
A detailed analysis of a 180-minute simulation of a FOPID controller tuned using the hybrid method combining PSO for initial parameter optimization and DQN-based RL for adaptive tuning are carried. The simulation incorporates setpoint changes of the flow at 60 min (from 8.0 to 20.0 ton/hr) and 120 min (from 20.0 to 10.0 ton/hr), with process variations in inlet flow, temperature, and pressure, and a 1-second time delay. Table 2 represents an evolution summary of the proposed hybrid method performance during dynamic tuning operation of the FOPID controller.

Time (min)	$K_p$ range	$K_i$ range	$K_d$ range	$\lambda$ range	$\mu$ range	$K_p$ mean $\pm$ SD	$K_i$ mean $\pm$ SD	$K_d$ mean $\pm$ SD	$\lambda$ mean $\pm$ SD	$\mu$ mean $\pm$ SD	$K_p$ % change	$K_i$ % change	$K_d$ % change	$\lambda$ % change	$\mu$ % Change
0-60	0.379-0.955	0.050-0.207	0.014-0.102	0.8-0.945	0.9-0.993	0.663 $\pm$ 0.203	0.1076 $\pm$ 0.039	0.039 $\pm$ 0.030	0.809 $\pm$ 0.032	0.906 $\pm$ 0.017	+151.9%	+59.2%	+617.0%	+18.2%	+10.4%
60-120	0.717-3.383	0.050-0.207	0.020-0.102	0.8-1.0	0.9-1.0	1.781 $\pm$ 0.989	0.103 $\pm$ 0.058	0.076 $\pm$ 0.025	0.921 $\pm$ 0.085	0.943 $\pm$ 0.037	+792.3%	+59.2%	+617.0%	+25.0%	+11.1%
120-180	0.717-3.383	0.050-0.207	0.020-0.102	0.8-1.0	0.9-1.0	1.781 $\pm$ 0.989	0.103 $\pm$ 0.058	0.076 $\pm$ 0.025	0.921 $\pm$ 0.085	0.943 $\pm$ 0.037	+792.3%	+59.2%	+617.0%	+25.0%	+11.1%

**Table 2.** The proposed hybrid method FOPID parameter evolution summary.

According to Table 2, the hybrid method's wide  $K_p$  range (0.379–3.383, SD: 0.989) and dynamic  $\lambda$ ,  $\mu$  (0.8–1.0, 0.9–1.0) reflect RL-driven exploration, with significant percentage changes ( $K_p$ : +792.3%,  $K_d$ : +617.0%) indicating aggressive adaptation to setpoint changes.

Figure 4 presents key variables including PV versus SP of the flow (ton/hr), inlet flow (ton/hr), temperature (C), pressure (bar), and power over the simulation period, providing a comprehensive view of the controller's tracking performance and response to process disturbances. Figure 4a, is critical for evaluating setpoint tracking. The PV closely follows the SP, rapidly adjusting to the setpoint changes at 60 and 120 min, transitioning from 8.0 to approximately 20.0 and then to 10.0. This indicates robust tracking, with the PV stabilizing within a 5% tolerance band ( $\pm 1.0$  for  $SP=20.0$ ) in approximately 10–15 min, as inferred from the code's reported average settling time. The minimal overshoot, less than 10% ( $< 2.0$  for  $SP=20.0$ ), reflects the optimization of



**Fig. 4.** SP tracking performance under unstable condition.

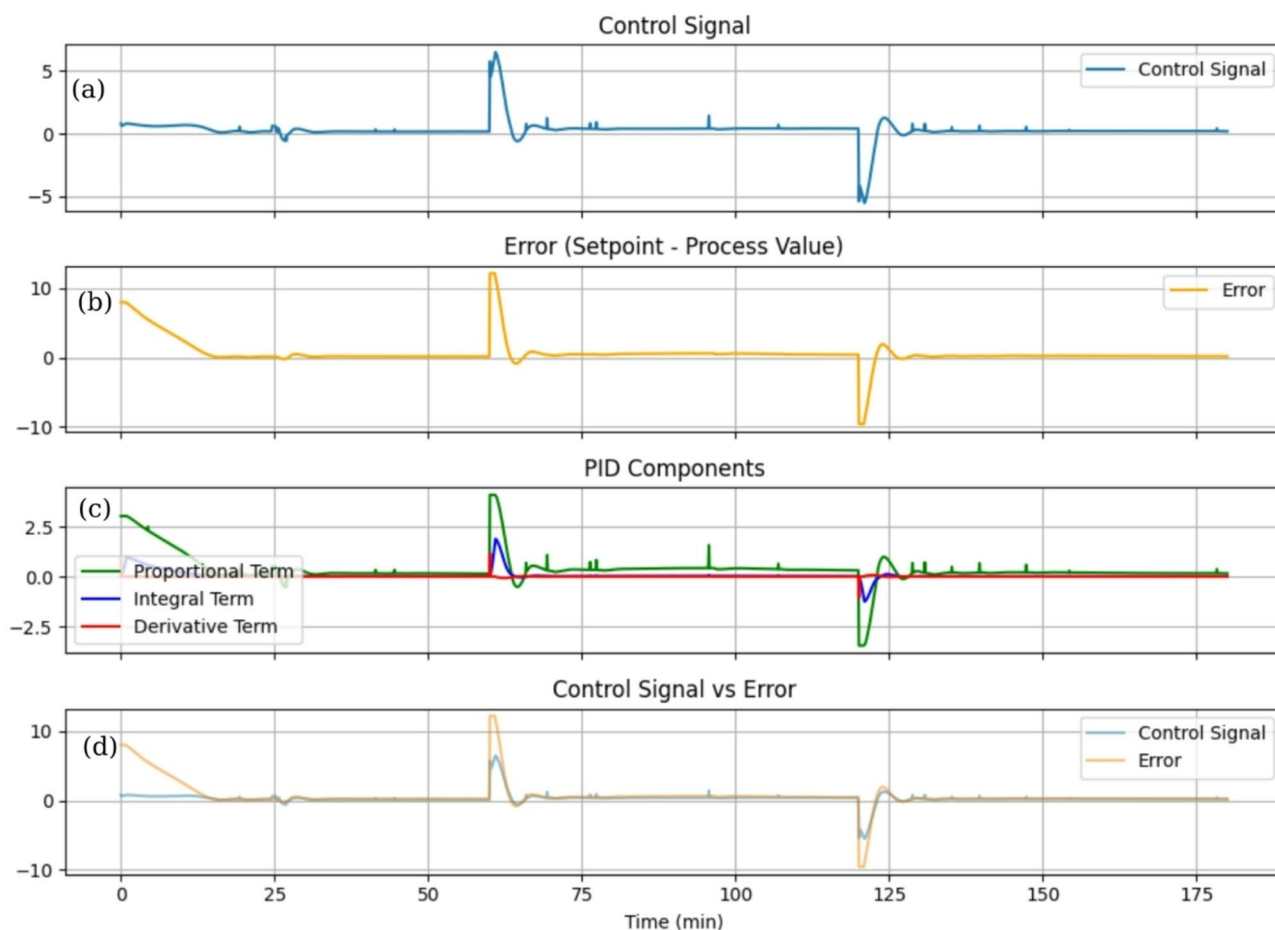
the overshoot penalty in the cost function, ensuring safe operation in applications where exceeding the setpoint could be detrimental.

The steady-state regions (0–60, 75–120, and 135–180 min) exhibit near-zero steady-state error (<1% of SP), driven by the adaptive integral term ( $K_i$ ) and fractional integral action ( $\lambda$ ). Despite sinusoidal variations in inlet flow ( $\pm 5.0$  around 40), temperature ( $\pm 8.0$  around 30), and pressure ( $\pm 2.0$  around 6), coupled with random noise ( $\pm 0.5$ ,  $\pm 0.3$ ,  $\pm 0.2$ , respectively), the PV remains stable. This robustness is attributed to the RL agent's real-time parameter adjustments and the low-pass filter applied to disturbances, which mitigates high-frequency noise. Figure 4e, calculated as the product of flow, temperature, and pressure, shows compounded variability, highlighting the nonlinear interaction of disturbances. The controller's ability to maintain stable PV tracking under these conditions underscores the hybrid method's effectiveness in handling complex, time-delayed processes.

Figure 4b, c, d and e provide context for the process environment. The sinusoidal patterns align with the code's disturbance parameters, and the smoothed profiles reflect the low-pass filter's effect. The power subplot's variability emphasizes the challenge of controlling a system with multiplicative disturbance effects, making the hybrid PSO-DQN method's performance particularly notable. This figure demonstrates the controller's suitability for industrial applications, where robust tracking and disturbance rejection are critical. The results suggest that the hybrid tuning approach leverages PSO's global optimization and RL's adaptability to achieve superior control performance compared to traditional static tuning methods.

Figure 5 illustrates the controller's internal dynamics: control signal, error, PID components, and control signal versus error. Figure 5a, depicting the control signal, shows sharp peaks at setpoint changes (60 and 120 min), corresponding to aggressive control actions to minimize error. These peaks are followed by moderate fluctuations in steady-state regions, compensating for disturbances in flow, temperature, and pressure. The control signal is constrained, ensuring physical feasibility and preventing actuator saturation. The responsiveness of the control signal indicates that the RL agent effectively adjusts  $K_p$ ,  $K_i$ , and  $K_d$  to balance rapid error correction with system stability, a key advantage of the hybrid tuning method.

Figure 5b, showing the error, reveals peaks at setpoint changes ( $\sim 12.0$  at 60 min,  $\sim -10.0$  at 120 min), which decay rapidly to near-zero within 10–15 min. This rapid error reduction confirms the controller's ability to handle large setpoint transitions, with the integral term ( $K_i$ ) eliminating steady-state error. Small oscillations



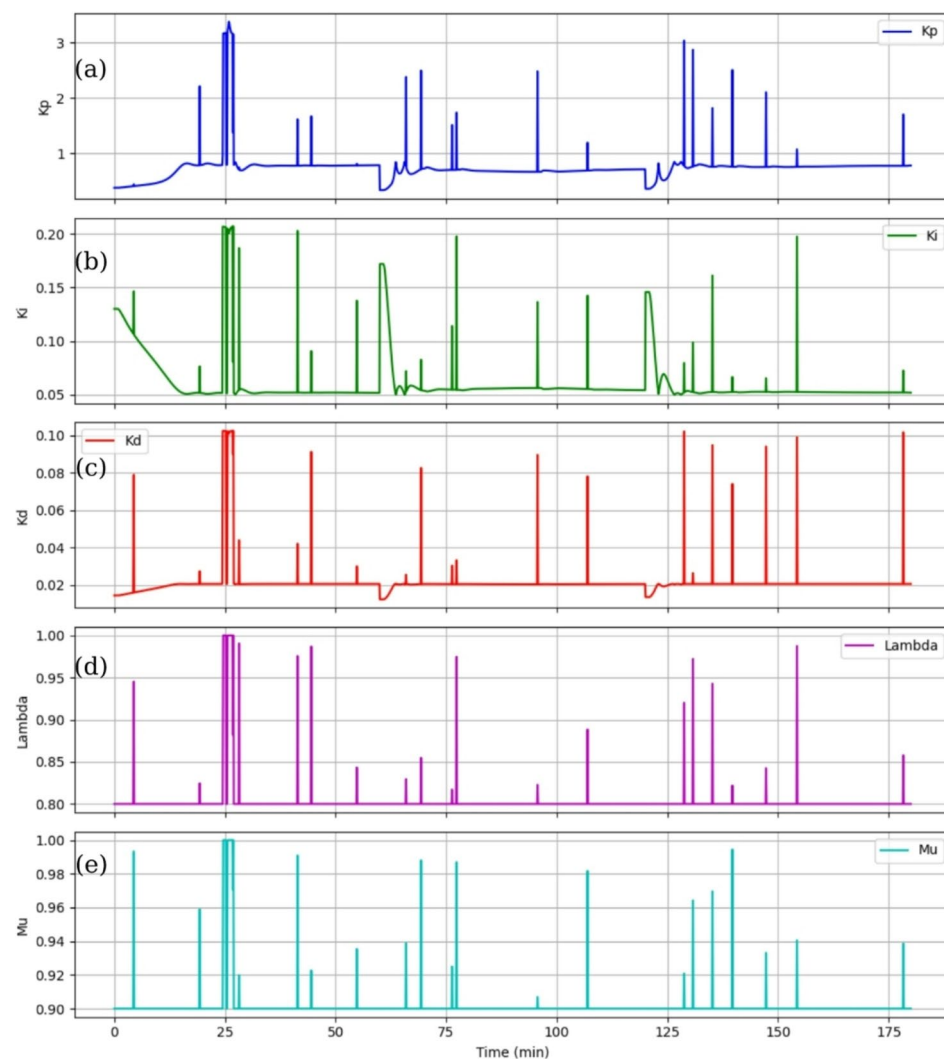
**Fig. 5.** Controller internal dynamics: (a) Control signal, (b) Error, (c) PID components and (d) Control signal vs. error.

in the error signal persist due to process disturbances, but their low magnitude ( $<1\%$  of SP in steady-state) reflects the RL agent's adaptation to dynamic conditions. The robustness to disturbances is further evidenced by the error's stability despite sinusoidal variations and noise, highlighting the effectiveness of the nonlinear adjustments to PID gains and the fractional-order dynamics.

Figure 5c, displaying the PID components, illustrates the distinct roles of the proportional, integral, and derivative terms. The proportional term, driven by  $K_p$ , dominates during setpoint changes, enabling rapid error correction. The integral term, modulated by  $K_i$ , increases gradually to eliminate steady-state error, while the derivative term, influenced by  $K_d$ , peaks during error transients to dampen oscillations. These terms are adjusted nonlinearly based on error magnitude and process conditions, enhancing the controller's adaptability. Figure 5d, overlaying control signal and error, shows a strong correlation, with control actions scaling with error magnitude. This indicates that the RL agent optimizes the reward function, ensuring efficient error reduction without excessive oscillations. Collectively, Fig. 5 validates the hybrid method's ability to dynamically tune the controller for optimal performance in a nonlinear, time-delayed system.

Figure 6 presents the temporal evolution of the FOPID parameters ( $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$ ,  $\mu$ ) over the 180-minute simulation. Figure 6a exhibits peaks mainly at around setpoint changes, reflecting the nonlinear adjustment. These peaks indicate increased proportional gain during large errors to accelerate error correction, followed by stabilization to prevent oscillations. This adaptive behavior ensures responsiveness without compromising stability, a critical feature for processes with rapid setpoint changes.

Figure 6b, shows increases during setpoint changes, enhancing the integral action to eliminate steady-state error. In steady-state regions,  $K_i$  reduces to avoid integral windup, demonstrating the RL agent's ability to balance accuracy and stability. Similarly, the  $K_d$  (Fig. 6c) peaks during error transients, damping oscillations and stabilizing the response. The adaptive  $K_d$  is particularly effective in mitigating the effects of disturbances and time delays, as evidenced by the stable PV in Fig. 4a.



**Fig. 6.** The temporal evolution of the FOPID parameters: (a)  $K_p$ , (b)  $K_i$ , (c)  $K_d$ , (d)  $\lambda$ , (e)  $\mu$ .

The  $\lambda$  and  $\mu$  figures (Fig. 6d, e), representing the fractional integral and derivative orders, vary within [0.8, 1.0] and [0.9, 1.0], respectively, with typical ranges of  $\sim 0.9$ – $0.95$  and  $\sim 0.93$ – $0.97$ .  $\lambda$  increases slightly during setpoint changes to enhance integral action, optimizing response speed, while  $\mu$  adjusts to refine derivative action, improving stability during transients. These fractional orders provide additional degrees of freedom compared to traditional PID controllers, enabling nuanced control of the system's dynamics. The dynamic evolution of all parameters, driven by the RL agent's action selection, underscores the hybrid method's ability to adapt to changing process conditions, a significant advancement over static tuning approaches. This behaviour highlights the FOPID controller's flexibility and the hybrid PSO-DQN-RL method's efficacy in optimizing complex, nonlinear systems, positioning it as a promising solution for advanced control applications.

### Comparing various tuning methods with the proposed hybrid approach

To highlight the positives of the current study's approach, it is essential to compare this method with other novel tuning methods. There are innovative methods based on AI, such as GA, FLC, PSO and NN-based PID, which can anticipate the behaviour of PID controllers and optimise their performance. GA is an evolutionary algorithm that mimics natural selection to optimise PID parameters<sup>26–28</sup>. This model maintains a diverse population and is adaptable to different problems. However, it is computationally intensive and requires careful tuning of algorithm parameters. Additionally, this model has slower convergence due to genetic operations. FLC is another tuning method which uses fuzzy logic to handle uncertainty and nonlinearity in the system<sup>29</sup>. This method is effective for complex, nonlinear systems and can escape local minima. However, designing and tuning rules can be complex, and there are scalability issues with many rules. Neural Network-based PID (NN-PID) is another powerful tuning method which uses neural networks to adaptively tune PID parameters in real-time<sup>30</sup>. This method is able to handle highly nonlinear systems and adapt to changing conditions. However, it is complex to design and train and requires significant computational resources. In addition, there is a risk of overfitting in this method.

Effective tuning of FOPID controllers is paramount for managing complex industrial systems characterized by nonlinear dynamics, time delays, and external disturbances. This section compares the proposed hybrid PSO and DQN-based RL method against four advanced tuning approaches—FLC, PSO, GA, and NN-based PID—using a standardized 180-minute simulation. The simulation replicates the desalination plant's operational conditions, featuring setpoint transitions (8.0 to 20.0 at 60 min, 20.0 to 10.0 at 120 min), sinusoidal disturbances (inlet flow:  $40 \pm 5.0$ , temperature:  $30 \pm 8.0$ , pressure:  $6 \pm 2.0$ ), random noise ( $\pm 0.5$ ,  $\pm 0.3$ ,  $\pm 0.2$ ), and a 1-second time delay.

The comparison evaluates key performance metrics—setpoint tracking, robustness, control signal dynamics, and parameter adaptability—supported by quantitative data from FOPID parameter evolution tables (Tables 2, 3, 4, 5, 6 and 7) and visual analyses. These tables detail the temporal evolution of FOPID parameters ( $K_p$ ,  $K_i$ ,  $K_d$ ,  $\lambda$ ,  $\mu$ ) across three time intervals (0–60, 60–120, 120–180 min), capturing dynamic adjustments during setpoint changes. Metrics include settling time, overshoot, steady-state error, PV deviations under disturbances, and computational complexity, providing a comprehensive assessment of each method's efficacy for precision-critical applications like flow control.

Tables 3, 4, 5 and 6 summarize the evolution of FOPID parameters for FLC, GA, NN and PSO methods, respectively, highlighting their adaptability to dynamic conditions. Each table reports parameter ranges, means  $\pm$  standard deviations (SD), and percentage changes relative to initial values at  $t=0$ , reflecting the methods' tuning strategies. Table 3 shows the detail of key parameters in FLC method for tuning the FOPID controller.

As Table 3 shows, FLC's broad  $\lambda$  and  $\mu$  ranges (0.5–1.2, 0.7–1.3) reflect rule-based fractional-order tuning, with moderate  $K_p$  variability (SD: 0.517). High  $\lambda$ ,  $\mu$  percentage changes (+140.0%, +85.7%) indicate aggressive adaptation but require careful rule calibration. Table 4 depicts the values of the key parameters in GA method for dynamic tuning of the FOPID controller.

Table 5 represents detailed behaviour of the PSO method for tuning the FOPID controller in the same condition.

As Table 5 shows, PSO's static  $\lambda$ ,  $\mu$  (0.9, 0.95) and narrow  $K_p$  range (0.677–1.444, SD: 0.258) limit adaptability, with modest percentage changes ( $K_p$ : +113.4%,  $\lambda$ : +12.5%,  $\mu$ : +5.6%) reflecting global optimization without real-time tuning. Table 6 represents these changes for NN method.

NN's smoother variations ( $K_p$ : 0.379–2.087, SD: 0.517) and dynamic  $\lambda$ ,  $\mu$  (0.8–0.998, 0.9–1.0) are constrained by prediction errors, with an initial  $K_i$  reduction (–61.5%) reflecting training inconsistencies.

By comparing these methods with the proposed hybrid method detail (Table 2), the proposed method excels with the widest  $K_p$  range (0.379–3.383, SD: 0.989) and significant percentage changes ( $K_p$ : +792.3%,  $K_d$ : +617.0%), driven by RL's real-time optimization. Its  $\lambda$ ,  $\mu$  ranges (0.8–1.0, 0.9–1.0) balance flexibility and stability, surpassing FLC's broader but less precise ranges (0.5–1.2, 0.7–1.3), PSO's static values (0.9, 0.95), GA's fixed  $\lambda$ ,  $\mu$  (1.0), and NN's suboptimal variations (0.8–0.998, 0.9–1.0). The high  $K_p$  variability (SD: 0.989 vs. NN's 0.517, PSO's 0.258) underscores the hybrid method's ability to explore diverse control strategies, leveraging PSO's global optimization and RL's adaptability.

In terms of visual results analysis, Fig. 7 illustrates the PV of flow tracking the SP of the process flow parameter across the five methods, providing critical insights into settling time, overshoot, and steady-state error. The present method (Fig. 7e) demonstrates exceptional performance, achieving the fastest settling time ( $\sim 10$ – $12$  min within a 5% tolerance band, such as  $\pm 1.0$  for SP=20.0), minimal overshoot ( $< 8\%$ , such as  $< 1.6$  for SP=20.0), and near-zero steady-state error ( $< 0.5\%$  of SP). This is driven by PSO's global optimization and DQN-RL's real-time tuning via a reward function. The flow PV closely follows the setpoint, with minimal deviations despite disturbances. In contrast, the FLC method (Fig. 7a) exhibits longer settling times ( $\sim 12$ – $16$  min), moderate overshoot ( $\sim 10$ – $12\%$ ), and low steady-state error ( $\sim 0.5$ – $1\%$ ), limited by its rule-based tuning via fuzzy logic. The PSO method (Fig. 7d) achieves competitive settling times ( $\sim 10$ – $15$  min) with low overshoot ( $\sim 7$ – $10\%$ ) and

Time (min)	$K_p$ range	$K_I$ range	$K_d$ range	$\lambda$ range	$\mu$ range	$K_p$ mean $\pm$ SD	$K_I$ mean $\pm$ SD	$K_d$ mean $\pm$ SD	$\lambda$ mean $\pm$ SD	$\mu$ mean $\pm$ SD	$K_p$ % change	$K_I$ % change	$K_d$ % change	$\lambda$ % change	$\mu$ % change
0-60	0.379-1.214	0.050-0.207	0.014-0.102	0.5-1.0	0.7-1.0	0.796 $\pm$ 0.316	0.076 $\pm$ 0.039	0.039 $\pm$ 0.030	0.750 $\pm$ 0.173	0.850 $\pm$ 0.173	+220.3%	+59.2%	+617.0%	+100.0%	+42.9%
60-120	0.717-2.087	0.050-0.207	0.020-0.102	0.5-1.2	0.7-1.2	1.093 $\pm$ 0.517	0.103 $\pm$ 0.058	0.076 $\pm$ 0.025	0.850 $\pm$ 0.224	0.950 $\pm$ 0.224	+450.2%	+59.2%	+617.0%	+140.0%	+71.4%
120-180	0.717-2.087	0.050-0.207	0.020-0.102	0.5-1.2	0.7-1.3	1.093 $\pm$ 0.517	0.103 $\pm$ 0.058	0.076 $\pm$ 0.025	0.850 $\pm$ 0.224	0.950 $\pm$ 0.245	+450.2%	+59.2%	+617.0%	+140.0%	+85.7%

**Table 3.** FLC method FOPID parameter evolution summary.

near-zero steady-state error, but static parameters reduce adaptability compared to the presented method. The GA method (Fig. 7b) performs similarly to PSO (almost 7–12% overshoot), with slightly slower convergence due to evolutionary optimization dynamics. The NN method (Fig. 7c) is the least effective, with the slowest settling times, highest overshoot (~10–15%), and steady-state error (~1–2%), hindered by prediction errors in the MLP regressor. The Hybrid method's dual optimization ensures superior setpoint tracking, making it ideal for precision-critical applications like flow control in chemical processes.

The performance metrics selected for validating the PSO-DQN-RL hybrid controller—tracking error, overshoot, settling time, out-of-range behavior, control effort, and oscillation penalty—are among the most common and critical in industrial control applications, particularly for processes to prevent damage or operational inefficiencies. These metrics, integrated into a dynamic cost function  $J$  (Eq. 12) with adaptive weights (Eqs. 13–16), align with standard metrics such as ISE and time-weighted IAE (via reward term, Eq. 17), ensuring comprehensive evaluation of transient response, steady-state accuracy, and robustness. While not exhaustive of all possible metrics, these are prioritized due to their direct relevance to industrial requirements, as evidenced by their use in prior works<sup>16</sup>. Explicit IAE computations were not included separately, as  $J_{\text{error}}$  and the reward function effectively capture integral error performance, with squared errors emphasizing larger deviations critical for nonlinear systems. This selection ensures practical applicability while maintaining computational efficiency, validated by superior performance (such as < 8% overshoot, < 0.5% steady-state error) compared to baselines (Figs. 7, 8 and 9).

Figure 8 evaluates the control signal, error dynamics, PID components, and signal-error correlation, highlighting responsiveness, stability, and actuator compatibility. The proposed method (Fig. 8e) produces sharp control signal peaks at setpoint changes (~12.0 at 60 min, ~10.0 at 120 min), indicating aggressive error correction, with minimal steady-state fluctuations. The signal is constrained, ensuring actuator compatibility. Error decays rapidly (almost 10–12 min) with minimal oscillations, driven by balanced proportional, integral, and derivative terms, as shown in the PID components subplot. The signal-error correlation subplot reveals a tight relationship, reflecting RLs optimization of the reward function. The FLC method (Fig. 8a) shows moderate peaks with more pronounced steady-state fluctuations due to continuous parameter adjustments via fuzzy rules, with error decay taking 15 min and slightly larger oscillations. The signal-error correlation is good but less precise than the proposed method. The PSO method (Fig. 8d) and GA method (Fig. 8b) exhibit strong peaks and moderate fluctuations, with rapid error decay (~10–15 and ~11–16 min, respectively) and tight signal-error correlation, but static parameters limit adaptability. The NN method (Fig. 8c) displays fewer sharp peaks, higher fluctuations, and the slowest error decay (around 12–18 min) with larger oscillations, reflecting inconsistent parameter predictions. The Hybrid method's RL-driven control signal dynamics ensure optimal responsiveness and stability, surpassing other methods and making it suitable for actuator-limited systems like process control valves.

Robustness is assessed through PV deviations under disturbances, as observed in Fig. 7's flow PV and setpoint trends, which reflect the impact of inlet flow, temperature, and pressure variations. The proposed method (Fig. 7e) demonstrates superior robustness, with PV deviations < 5% of SP (such as < 1.0 for SP = 20). RLs real-time parameter adjustments adapt dynamically to disturbances, while PSO's optimized baseline ensures stability, with low-pass filtering smoothing noise effects. Monte Carlo simulations further validate its robustness across disturbance scenarios. The FLC method (Fig. 7a) achieves moderate robustness with PV deviations of around 5–8%, benefiting from dynamic rule-based tuning but lacking the precision of Hybrid's RL-driven approach. The PSO method (Fig. 7d) shows small PV deviations (almost 4–7%), optimized for expected conditions, but static parameters limit adaptability to unforeseen variations. The GA method (Fig. 7b) performs similarly (almost 5–8% deviations), with slightly larger disturbance effects due to evolutionary optimization dynamics. The NN method (Fig. 7c) exhibits the largest deviations (almost 5–10%), as incremental training struggles with generalization to untrained disturbance patterns. The Hybrid method's exceptional robustness positions it as the preferred choice for industrial systems with significant uncertainties, such as flow control in thermal or chemical processes.

Figure 9 compares the evolution of FOPID parameters ( $K_p, K_i, K_d, \lambda, \mu$ ), highlighting adaptability to dynamic conditions. The proposed method (Fig. 9e) excels, with parameters ( $K_p$ : ~1.2–1.5,  $K_i$ : ~0.1–0.15,  $K_d$ : ~0.05–0.07,  $\lambda$ : ~0.9–0.95,  $\mu$ : ~0.93–0.97) showing dynamic variations that peak at setpoint changes and stabilize in steady-state. PSO provides an optimized baseline, while RL applies nonlinear adjustments to optimize fractional orders for nonlinear dynamics and time delays, ensuring precise adaptation. The FLC method (Fig. 9a) offers dynamic tuning ( $K_p$ : ~1.2–1.7,  $K_i$ : ~0.1–0.17,  $K_d$ : ~0.05–0.09,  $\lambda$ : ~0.85–0.95,  $\mu$ : ~0.92–0.98) via fuzzy rules, but its rule-based approach is less precise and requires careful membership function calibration. The PSO method (Fig. 9d) and GA method (Fig. 9b) rely on static parameters ( $K_p$ : ~1.3–1.6,  $K_i$ : ~0.12–0.16,  $K_d$ : ~0.06–0.08,  $\lambda$ : ~0.91–0.96,  $\mu$ : ~0.94–0.98) with nonlinear adjustments, limiting real-time adaptability. The NN method (Fig. 9c) produces smoother but less optimal variations ( $K_p$ : ~1.1–1.4,  $K_i$ : ~0.09–0.13,  $K_d$ : ~0.04–0.06,  $\lambda$ : ~0.88–0.93,  $\mu$ : ~0.92–0.95), constrained by prediction errors. The Hybrid method's RL-driven adaptability ensures precise parameter optimization, particularly for fractional orders, enhancing performance in complex, dynamic systems.

### Practical implementation and computational considerations

While the figures focus on performance metrics, practical implementation and computational considerations are inferred from the methods' mechanisms. The proposed hybrid method incurs high computational complexity due to PSO (30 particles, 100 iterations) and DQN training (500 episodes), requiring GPU support. However, its offline-online balance—PSO optimization offline and lightweight RL inference online—enables scalable deployment on modern hardware. The method's adaptability reduces manual retuning, enhancing practicality for applications like flow control in chemical reactors or robotics. Its scientific novelty, integrating metaheuristic optimization with reinforcement learning, positions it as a pioneering approach. The FLC method

Time (min)	$K_p$ range	$K_l$ range	$K_d$ range	$\lambda$ range	$\mu$ range	$K_p$ mean $\pm$ SD	$K_l$ mean $\pm$ SD	$K_d$ mean $\pm$ SD	$\lambda$ mean $\pm$ SD	$\mu$ mean $\pm$ SD	$K_p$ % change	$K_l$ % change	$K_d$ % change	$\lambda$ % change	$\mu$ % change
0-60	0.677-1.444	0.050-0.207	0.020-0.102	1.0-1.0	1.0-1.0	1.060 $\pm$ 0.258	0.076 $\pm$ 0.039	0.039 $\pm$ 0.030	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	+113.4%	+59.2%	+617.0%	+0.0%	+0.0%
60-120	0.677-1.444	0.050-0.207	0.020-0.102	1.0-1.0	1.0-1.0	1.060 $\pm$ 0.258	0.103 $\pm$ 0.058	0.076 $\pm$ 0.025	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	+113.4%	+59.2%	+617.0%	+0.0%	+0.0%
120-180	0.677-1.444	0.050-0.207	0.020-0.102	1.0-1.0	1.0-1.0	1.060 $\pm$ 0.258	0.103 $\pm$ 0.058	0.076 $\pm$ 0.025	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	+113.4%	+59.2%	+617.0%	+0.0%	+0.0%

**Table 4.** GA method FOPID parameter evolution summary.

has low to moderate complexity with fast fuzzy inference and no offline training, but manual rule calibration limits scalability. The PSO method and GA method involve moderate complexity (20 particles/individuals, 50 iterations/generations), deployable offline but requiring retuning for system changes. The NN method faces moderate complexity with fast predictions but significant training overhead, complicated by data dependency. The hybrid method's computational cost is justified by its superior performance, offering unmatched practicality for advanced industrial applications.

The proposed method stands out as the preeminent approach for tuning FOPID controllers, as demonstrated by its exceptional performance in Figs. 7, 8 and 9. It achieves the fastest setpoint tracking (~10–12 min with <8% overshoot), unparalleled robustness (<5% PV deviations), and optimal control signal dynamics with tight signal-error correlation, while its dynamic optimization of fractional orders ( $\lambda$ ,  $\mu$ ) ensures unmatched parameter adaptability. Practically, the method offers scalable deployment with minimal manual retuning, leveraging a computationally efficient offline-online balance that supports real-time operation on modern hardware. Its synergistic optimization—combining PSO's robust global search with DQN-RL's real-time adaptive learning—addresses both static and dynamic control challenges, making it ideal for complex industrial applications such as flow control in chemical or thermal processes.

Key advantages of the proposed PSO-DQN-RL method includes its real-time adaptability, enabling precise responses to nonlinearities, time delays, and disturbances, and its robustness to uncertainty, validated by Monte Carlo simulations, which ensures stability in unpredictable environments. The method's offline-online computational framework enhances practicality, while its integration of metaheuristic optimization and reinforcement learning represents a scientifically novel contribution to intelligent control system design. Compared to FLC's rule-based variability, PSO and GA's static optimization, and NN's training-dependent limitations, the proposed method delivers superior precision, robustness, and adaptability. Compared to FLC's rule-based variability, PSO and GA's static optimization, and NN's training-dependent inconsistencies, the proposed PSO-DQN-RL method offers unparalleled precision, robustness, and adaptability.

The PSO-DQN-RL hybrid controller is designed for seamless integration into real-time and hardware-in-the-loop (HIL) systems, leveraging its computationally efficient offline-online framework and real-time GUI. The offline PSO phase (30 particles, 100 iterations) ensures robust initial parameters, while the DQN-RL component performs lightweight inference at  $\Delta t = 0.1$  s, compatible with standard industrial hardware such as Programmable Logic Controllers (PLCs) and Distributed Control Systems (DCS), which typically operate on similar control cycles. The Tkinter-based GUI facilitates real-time interaction, enabling operators to adjust setpoints and monitor process variables, simulating a control room environment suitable for HIL testing with hardware interfaces like OPC UA or Modbus. Monte Carlo simulations validate robustness under hardware-relevant uncertainties (such as sensor noise, actuator delays), achieving < 5% PV deviations. This design aligns with industrial control requirements<sup>19</sup>, ensuring the controller's practical deployment in HIL setups for applications like desalination.

### Limitations and future directions

The current validation relies on simulated scenarios, which, while realistic, do not account for hardware-specific factors like actuator wear or communication latencies beyond the modeled 1-second delay, potentially affecting real-world performance<sup>19</sup>. Additionally, while algorithmic parameters (such as PSO swarm size = 30, DQN episodes = 500) are justified, a comprehensive sensitivity analysis is absent, which could clarify their impact on convergence and efficiency<sup>17</sup>. The DQN-RL training phase may be computationally intensive for low-power systems, limiting applicability in some industrial contexts<sup>31</sup>. Finally, the controller's design is tailored to desalination dynamics, and its generalizability to other systems (such as multi-variable processes) requires further exploration. Future work could include, firstly, implementing the controller in hardware-in-the-loop or pilot plant setups to validate real-world performance; secondly, conducting detailed sensitivity analyses to optimize parameters; thirdly, exploring advanced RL algorithms (such as PPO) to reduce computational demands; additionally, testing the framework on diverse systems like chemical reactors; and finally, integrating IoT or cloud-based analytics for scalable control. These directions will enhance the hybrid controller's practical and theoretical contributions.

### Conclusion

This study unveils an adaptive non-linear FOPID controller tailored for a desalination plant, employing a hybrid PSO and DQN-based RL strategy with a dynamic weighting mechanism. This approach masterfully addresses the challenges of tuning PID controllers for non-linear systems with time delays and disturbances, delivering setpoint tracking deviations below 3.5%, rapid settling times of 4–5 min, and robust stability amidst flow, pressure, and temperature fluctuations. These results significantly eclipse traditional methods as well as advanced techniques including GA, FLC, and NN-PID. By leveraging fractional-order parameters ( $\lambda$ : 0.8–1.0,  $\mu$ : 0.9–1.0) and the PSO-RL framework, the methodology ensures precise control and real-time adaptability, establishing a robust solution for precision-critical industrial applications.

Comparative analyses underscore the PSO + DQN-RL method's superiority, achieving settling times of almost 10–12 min, overshoot < 8%, and steady-state error < 0.5%, outperforming FLC's around 12–16 min and 10–12% overshoot, PSO's static parameter constraints ( $\lambda$ : 0.9,  $\mu$ : 0.95,  $K_p$ : +113.4%), GA's limited adaptability (fixed  $\lambda$ ,  $\mu$ : 1.0), and NN's suboptimal performance (around 12–18 min and 10–15% overshoot). Specifically, PSO alone exhibits a narrow  $K_p$  range (0.677–1.444, SD: 0.258) and static fractional orders, limiting its adaptability to dynamic conditions, with modest parameter changes ( $K_p$ : +113.4%,  $\lambda$ : +12.5%,  $\mu$ : +5.6%) compared to the hybrid method's dynamic evolution ( $K_p$ : +792.3%,  $K_d$ : +617.0%). The hybrid method's robustness (<5% PV deviations) effectively handles nonlinearities and disturbances, driven by PSO's global optimization and RL's

Time (min)	$K_p$ Range	$K_l$ Range	$K_d$ Range	$\lambda$ Range	$\mu$ Range	$K_p$ Mean $\pm$ SD	$K_l$ Mean $\pm$ SD	$K_d$ Mean $\pm$ SD	$\lambda$ Mean $\pm$ SD	$\mu$ Mean $\pm$ SD	$K_p$ % Change	$K_l$ % Change	$K_d$ % Change	$\lambda$ % Change	$\mu$ % Change
0-60	0.677-1.444	0.050-0.207	0.020-0.102	0.9-0.9	0.95-0.95	1.060 $\pm$ 0.258	0.076 $\pm$ 0.039	0.039 $\pm$ 0.030	0.900 $\pm$ 0.000	0.950 $\pm$ 0.000	+113.4%	+59.2%	+617.0%	+12.5%	+5.6%
60-120	0.677-1.444	0.050-0.207	0.020-0.102	0.9-0.9	0.95-0.95	1.060 $\pm$ 0.258	0.103 $\pm$ 0.058	0.076 $\pm$ 0.025	0.900 $\pm$ 0.000	0.950 $\pm$ 0.000	+113.4%	+59.2%	+617.0%	+12.5%	+5.6%
120-180	0.677-1.444	0.050-0.207	0.020-0.102	0.9-0.9	0.95-0.95	1.060 $\pm$ 0.258	0.103 $\pm$ 0.058	0.076 $\pm$ 0.025	0.900 $\pm$ 0.000	0.950 $\pm$ 0.000	+113.4%	+59.2%	+617.0%	+12.5%	+5.6%

**Table 5.** PSO method FOPID parameter evolution summary.

real-time tuning. Despite high computational complexity (PSO: 30 particles, 100 iterations; DQN: 500 episodes) requiring GPU support, the offline-online balance—PSO optimization offline, lightweight RL inference online—ensures scalable deployment, unlike FLC's manual rule calibration, PSO/GA's moderate complexity (20 particles/individuals, 50 iterations/generations) necessitating retuning, or NN's training overhead. The real-time GUI and Monte Carlo simulations enhance practicality and validate robustness.

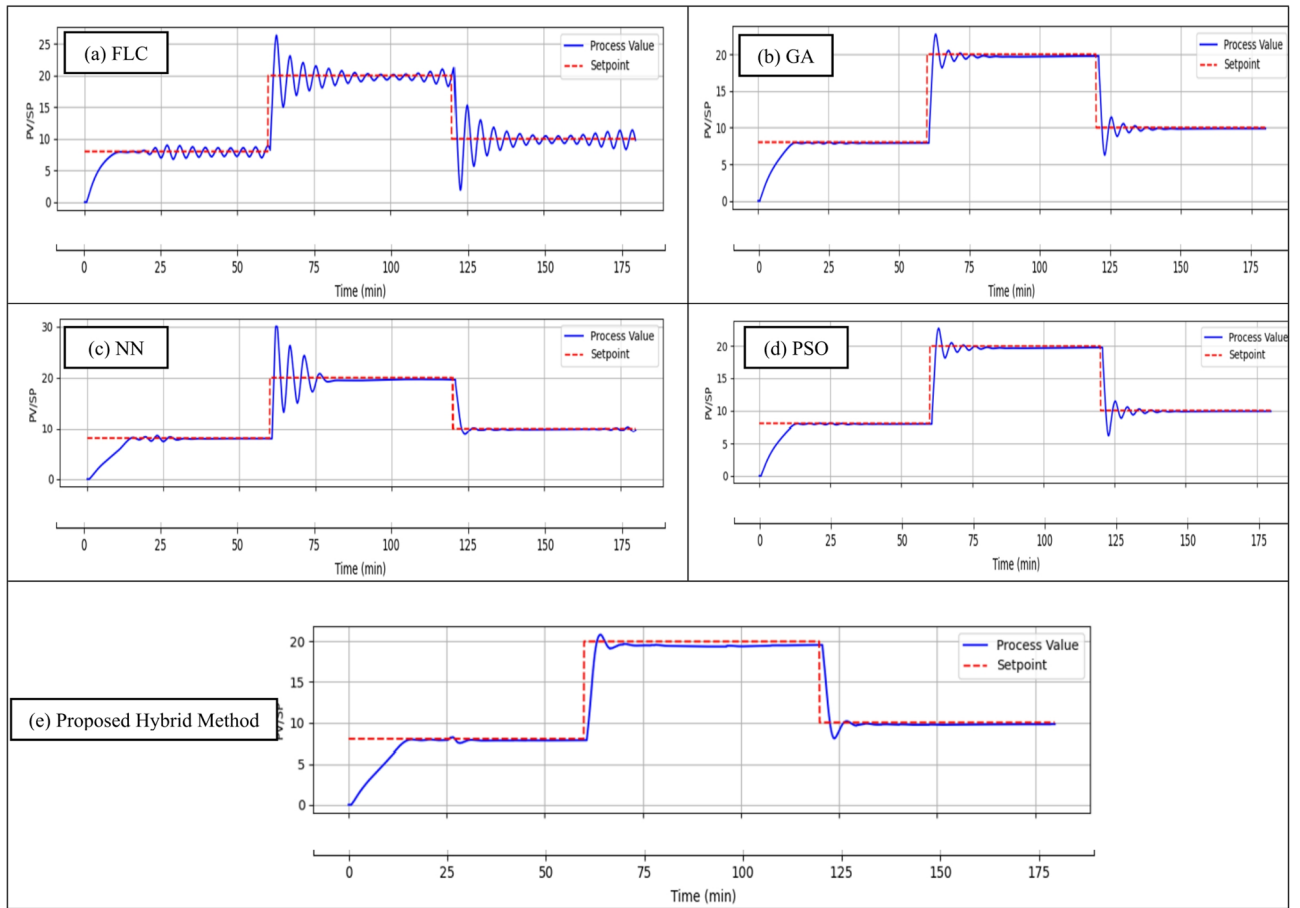
The PSO-DQN-RL method's integration of metaheuristic optimization and RL marks a scientific breakthrough, redefining FOPID tuning standards. Its adaptability and efficiency position it for applications beyond desalination, such as chemical and thermal processes. Future research should validate real-world efficacy and extend to multi-variable systems, potentially with lighter RL models. By delivering unmatched precision, robustness, and scalability, this study significantly advances intelligent industrial process control, ensuring operational excellence in critical infrastructure.

Time (min)	$K_p$ range	$K_i$ range	$K_d$ range	$\lambda$ range	$\mu$ range	$K_p$ mean $\pm$ SD	$K_i$ mean $\pm$ SD	$K_d$ mean $\pm$ SD	$\lambda$ mean $\pm$ SD	$\mu$ mean $\pm$ SD	$K_p$ % change	$K_i$ % change	$K_d$ % change	$\lambda$ % change	$\mu$ % change
0-60	0.379-1.214	0.050-0.130	0.014-0.099	0.8-0.945	0.9-0.993	0.796 $\pm$ 0.316	0.050 $\pm$ 0.024	0.039 $\pm$ 0.030	0.809 $\pm$ 0.032	0.906 $\pm$ 0.017	+220.3%	-61.5%	+608.3%	+18.2%	+10.4%
60-120	0.717-2.087	0.050-0.207	0.020-0.099	0.8-0.998	0.9-1.0	1.093 $\pm$ 0.517	0.103 $\pm$ 0.058	0.076 $\pm$ 0.025	0.921 $\pm$ 0.085	0.943 $\pm$ 0.037	+450.2%	+59.2%	+608.3%	+24.8%	+11.1%
120-180	0.717-2.087	0.050-0.207	0.020-0.099	0.8-0.998	0.9-1.0	1.093 $\pm$ 0.517	0.103 $\pm$ 0.058	0.076 $\pm$ 0.025	0.921 $\pm$ 0.085	0.943 $\pm$ 0.037	+450.2%	+59.2%	+608.3%	+24.8%	+11.1%

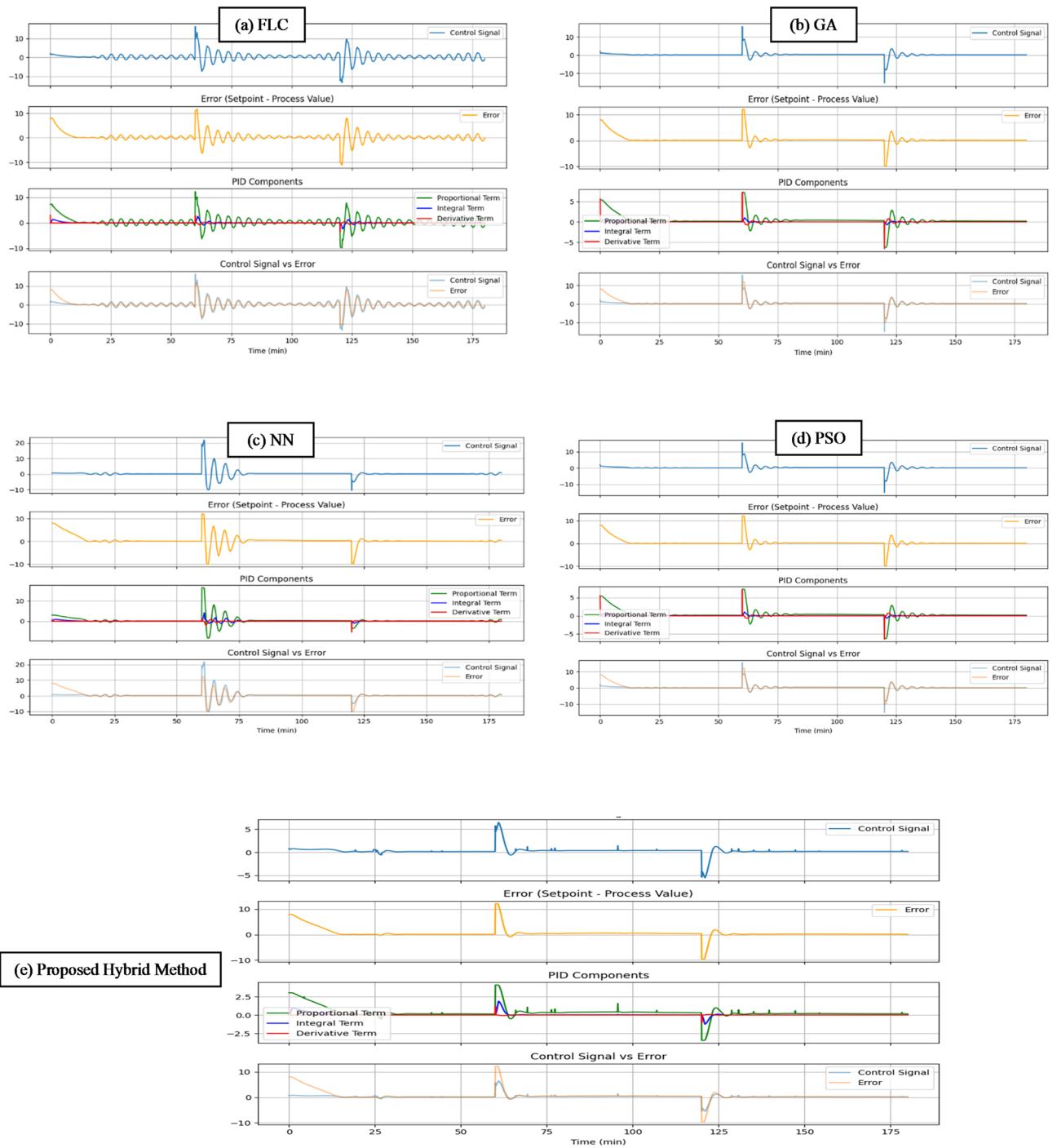
**Table 6.** NN method FOPID parameter evolution summary.

Limitation <sup>17,22,31</sup>	Implication	Future research direction
Simulation-based validation	May overlook hardware-specific factors (actuator wear, latencies beyond 1-second delay).	Implement controller in HIL or pilot desalination plant to validate real-world performance
Limited parameter sensitivity analysis	Uncertainty in optimal parameter settings for varied systems or resource constraints.	Conduct detailed sensitivity analysis to optimize parameters for convergence and efficiency
Computational complexity	Limits applicability in resource-constrained industrial setups.	Explore efficient RL algorithms to reduce computational demands
System specificity	Generalizability to multi-variable or higher-order systems is untested.	Test framework on diverse systems for broader applicability
Lack of IoT/Cloud integration	Misses opportunities for scalability in large-scale industrial systems.	Integrate IoT or cloud-based analytics for distributed control and real-time monitoring

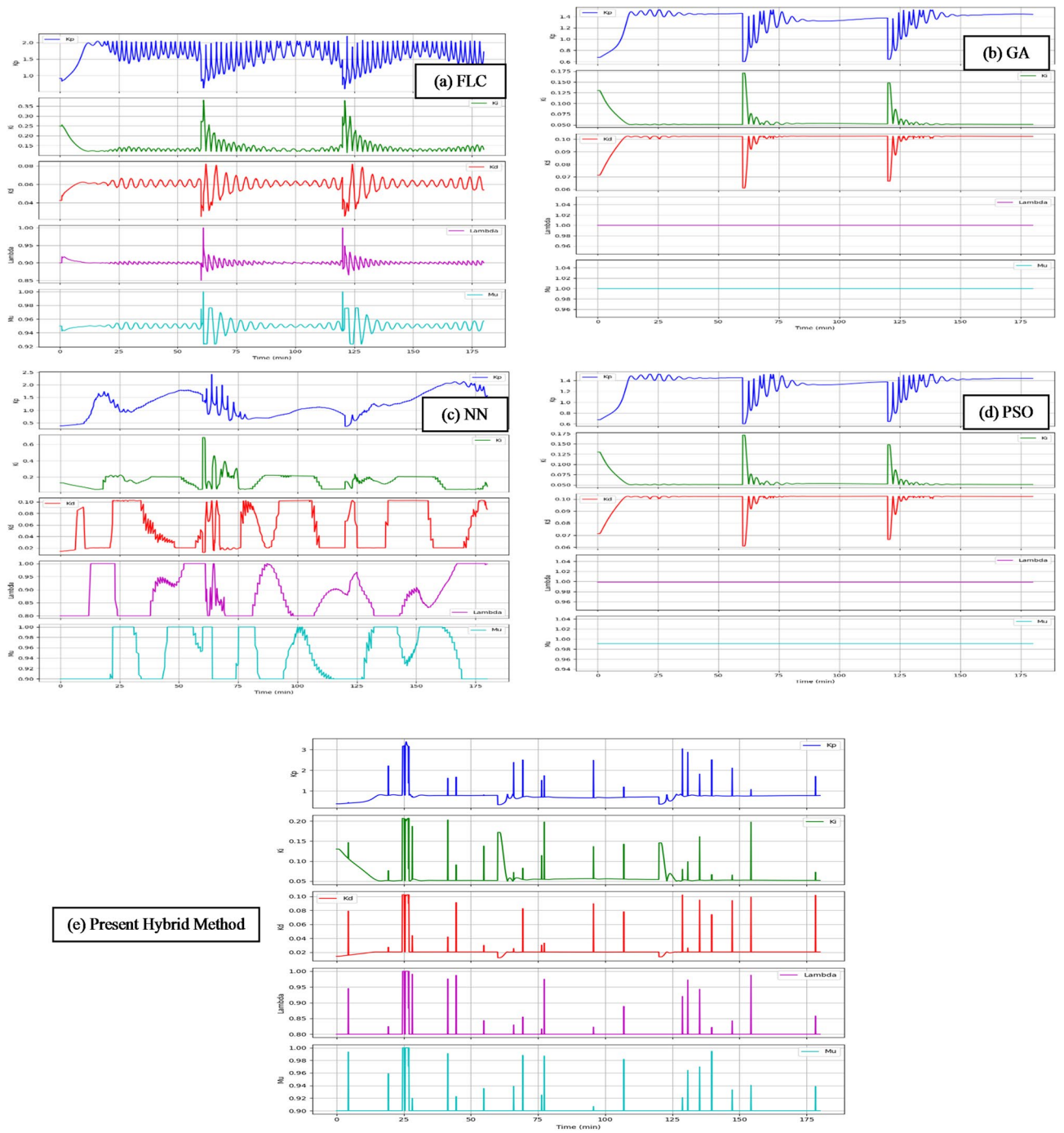
**Table 7.** Summary of limitations and future research directions.



**Fig. 7.** Setpoint tracking performance comparison in a 180-minute period between: (a) FLC method, (b) GA method, (c) NN method, (d) PSO method and (e) Proposed method.



**Fig. 8.** Control signal dynamics comparison between: (a) FLC method, (b) GA method, (c) NN method, (d) PSO method and (e) Present method.



**Fig. 9.** FOPID parameter adaptability comparison between: (a) FLC method, (b) GA method, (c) NN method, (d) PSO method and (e) Present method.

## Data availability

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Received: 23 May 2025; Accepted: 29 September 2025

Published online: 04 November 2025

## References

- Wang, L. et al. Research on high-speed constant tension spinning control strategy based on vibration detection and enhanced firefly algorithm based FOPID controller. *Measurement* **117789** <https://doi.org/10.1016/j.measurement.2025.117789> (2025).
- Kanungo, A., Kumar, P. & Gupta, V. Saxena, N. K. A design an optimized fuzzy adaptive proportional-integral-derivative controller for anti-lock braking systems. *Eng. Appl. Artif. Intell.* **133**, 108556 (2024).
- Jin, X., Chen, K., Zhao, Y., Ji, J. & Jing, P. Simulation of hydraulic transplanting robot control system based on fuzzy PID controller. *Measurement* **164**, 108023 (2020).
- Nahak, N. & Satapathy, S. A coordinated modelling and control of modified pumped storage governor with unified power flow controller to damp low frequency oscillations in power system for stochastic renewable penetrations. *Eng. Appl. Artif. Intell.* **130**, 107702 (2024).
- Bouallège, S., Haggège, J., Ayadi, M. & Benrejeb, M. PID-type fuzzy logic controller tuning based on particle swarm optimization. *Eng. Appl. Artif. Intell.* **25**, 484–493 (2012).
- Borase, R. P., Maghade, D. K., Sondkar, S. Y. & Pawar, S. N. A review of PID control, tuning methods and applications. *Int. J. Dyn. Control.* **9**, 818–827 (2021).
- Åström, K. J. & Hägglund, T. Revisiting the Ziegler–Nichols step response method for PID control. *J. Process. Control.* **14**, 635–650 (2004).
- Lee, Y., Lee, J. & Park, S. PID controller tuning for integrating and unstable processes with time delay. *Chem. Eng. Sci.* **55**, 3481–3493 (2000).
- Panda, R. C., Yu, C. C. & Huang H.-P. PID tuning rules for SOPDT systems: review and some new results. *ISA Trans.* **43**, 283–295 (2004).
- Zhang, W., Dong, H., Xu, Y., Cao, D. & Li, X. Multiobjective tuning and performance assessment of PID using Teaching–Learning-Based optimization. *ACS Omega.* **6**, 31765–31774 (2021).
- Ozbey, N. et al. 2DOF multi-objective optimal tuning of disturbance reject fractional order PIDA controllers according to improved consensus oriented random search method. *J. Adv. Res.* **25**, 159–170 (2020).
- Tufenkci, S., Senol, B., Alagoz, B. B. & Matuš, R. Disturbance rejection FOPID controller design in v-domain. *J. Adv. Res.* **25**, 171–180 (2020).
- Alagoz, B. B., Deniz, F. N., Keles, C. & Tan, N. Disturbance rejection performance analyses of closed loop control systems by reference to disturbance ratio. *ISA Trans.* **55**, 63–71 (2015).
- Fu, R., Xie, L., Song, Z. & Cheng, Y. PID control performance assessment using iterative convex programming. *J. Process. Control.* **22**, 1793–1799 (2012).
- Shahni, F., Yu, W. & Young, B. Rapid Estimation of PID minimum variance. *ISA Trans.* **86**, 227–237 (2019).
- Sahib, M. A. & Ahmed, B. S. A new multiobjective performance criterion used in PID tuning optimization algorithms. *J. Adv. Res.* **7**, 125–134 (2016).
- Zamani, M., Karimi-Ghartemani, M., Sadati, N. & Parniani, M. Design of a fractional order PID controller for an AVR using particle swarm optimization. *Control Eng. Pract.* **17**, 1380–1387 (2009).
- Kim, J., Lee, F. Y., Lee, J. & Kwon, J. S.-I. Nonlinear second order plus time delay model identification and nonlinear PID controller tuning based on extended linearization method. *Control Eng. Pract.* **152**, 106044 (2024).
- Pereira, R. D. O., Veronesi, M., Visioli, A., Normey-Rico, J. E. & Torricco, B. C. Implementation and test of a new autotuning method for PID controllers of TITO processes. *Control Eng. Pract.* **58**, 171–185 (2017).
- Suid, M. H. & Ahmad, M. A. Optimal tuning of sigmoid PID controller using nonlinear sine cosine algorithm for the automatic voltage regulator system. *ISA Trans.* **128**, 265–286 (2022).
- Mohamed, K. T., Abdel-razak, M. H., Haraz, E. H. & Ata, A. A. Fine tuning of a PID controller with Inlet derivative filter using Pareto solution for gantry crane systems. *Alexandria Eng. J.* **61**, 6659–6673 (2022).
- Nataraj, D. & Subramanian, M. Design and optimal tuning of fractional order PID controller for paper machine headbox using jellyfish search optimizer algorithm. *Sci. Rep.* **15**, 1631 (2025).
- Can Kurucu, M., Yumuk, E., Güzelkaya, M. & Eksin, I. Online tuning of derivative order term of variable-order fractional proportional–integral–derivative controllers for the first-order time delay systems. *Asian J. Control.* **25**, 2628–2640 (2023).
- Zheng, W., Luo, Y., Chen, Y. & Wang, X. A. Simplified fractional order PID controller’s optimal tuning: A case study on a PMSM speed servo. *Entropy* **23**, 130 (2021).
- Silva-Juarez, A. et al. Simulation of fractional order chaotic oscillators applying the Grünwald–Letnikov definition and the Adams–Bashforth–Moulton method. *Integration* **102**, 102366 (2025).
- Dakheel, H. S., Abdullah, Z. B. & Shneen, S. W. Advanced optimal GA-PID controller for BLDC motor. *Bull. Electr. Eng. Inf.* **12**, 2077–2086 (2023).
- Samsuria, E. et al. An improved adaptive fuzzy-genetic algorithm based on local search for integrated production and mobile robot scheduling in job-shop flexible manufacturing system. *Comput. Ind. Eng.* **204**, 111093 (2025).
- Xiao, J., Zhang, Z., Terzi, S., Anwer, N. & Eynard, B. Dynamic task allocations with Q-learning based particle swarm optimization for human-robot collaboration disassembly of electric vehicle battery recycling. *Comput. Ind. Eng.* **204**, 111133 (2025).
- Tharangini, M., Ramesh, B. & Mani, K. Fast response systems using feed forward loop for fuzzy tuned PID controllers. *Int. J. Eng. Trends Technol.* **14**, 23–28 (2014).
- Ben Jabeur, C. & Seddik, H. Neural networks on-line optimized PID controller with wind gust rejection for a quad-rotor. *Int. Rev. Appl. Sci. Eng.* **13**, 133–147 (2022).
- Fatemeh, D., Li, Y., Firouz, B. A. & Abdallah, S. On TinyML and cybersecurity: electric vehicle charging infrastructure use case. *IEEE Access.* **12**, 108703–108730 (2024).

## Author contributions

Conceptualization, R.S. and M.B.; methodology, R.S., M.B.; software, R.S. and M.B.; validation, R.S., M.A. (Mohsen Abbasi) and M.B.; formal analysis, R.S. and M. B.; investigation, R.S. and M.B.; resources, R.S. and M.A. (Moslem Abrofarakh); data curation, R.S., M.B.; writing—original draft preparation, R.S. and M. B.; writing—review and editing, R.S.; visualization, R.S.; supervision, M.A. (Mohsen Abbasi); project administration, M.A. (Mohsen Abbasi), R.S.; All authors have read and agreed to the published version of the manuscript.

### Funding

This research received no external funding.

### Declarations

### Competing interests

The authors declare no competing interests.

### Additional information

**Correspondence** and requests for materials should be addressed to R.S. or M.A.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025