Journal of Science and Technology – Smart Systems and Devices
Volume 36, Issue 2, May 2026, 057–065

DDPG-Based Intelligent Control Strategy for Industrial Dual-Tank System

Minh-Anh Nguyen, Tung-Lam Tran, Quoc-Dai Lai, Phuc-Lam Hoang, Thu-Ha Nguyen*
Hanoi University of Science and Technology, Ha Noi, Vietnam
**Corresponding author email: ha.nguyenthu3@hust.edu.vn*

Abstract
The performance and reliability of industrial control systems are impacted by external influences such as fluctuating operating environments and disruptive interferences, presenting notable challenges. Consequently, the exploration and adoption of smart control algorithms with capabilities for autonomous learning, self-tuning, and self-adjustment have emerged as a vital and significant research focus. This research investigates an intelligent control technique that employs the Deep Deterministic Policy Gradient (DDPG) algorithm for process control systems with a dual-tank system selected as the case study, in which the flow rate is manipulated to regulate the system temperature. The performance of the observer component in the critic network is improved by integrating a densely connected layer, which enhances its capacity to represent and handle data, thereby improving the identification of essential characteristics for water-level management. Additionally, the neural network's node settings are fine-tuned, and the ReLU activation function is implemented to support ongoing monitoring and adaptation to the external tank environment while preventing gradient vanishing. The research firstly trains DDPG with various initial conditions and then validates the performance for the temperature control problem by simulation. Additionally, the performance of DDPG is compared to the conventional Proportional-Integral-Derivative (PID) controller in terms of rise time, settling time, overshoot, and steady-state error.

Keywords: DDPG, dual-tank system, intelligent control technique, reinforcement learning.

1. Introduction

Industrial process control plays a critical role in ensuring the safe, efficient, and reliable operation of modern manufacturing and production systems. It encompasses the continuous measurement, monitoring, and adjustment of critical process variables to maintain stability, achieve desired outputs, and optimize overall system performance [1]. These variables include temperature, which is crucial for reactions in chemical plants, heat treatment in metallurgy, and sterilization in food processing; pressure, which ensures proper flow in pipelines, safe operation of boilers, and stability in high-pressure reactors; and flow rate, which governs the accurate delivery of liquids, gases, or steam in processes such as distillation, pumping, and cooling systems. In addition, level control in tanks and vessels is essential to prevent overflow, maintain proper feedstock supply, and ensure continuous production, while concentration and composition control guarantees product quality in processes like blending, fermentation, and refining. Other variables, such as pH, humidity, viscosity, and density, are critical in specialized industries like pharmaceuticals, food, and petrochemicals. By effectively managing these parameters in real time, industrial process control systems enable companies to improve energy efficiency, reduce material waste, enhance product consistency, and comply with stringent safety and environmental regulations.

In order to control process variables, the conventional Proportional-Integral-Derivative (PID) controller is widely used in both academic research and industrial applications. The research in [2] investigates theclosed-loop control with PID controller to maintain the flow of water in the process control system. This paper focuses on analyzing the parameter tuning based on the flow rate, the effectiveness of the PID controller is not made clear. The study in [3] improves the performance of the PID controller based on fuzzy logic for the dual-tank system. In particular, parameters of PID controller are adjusted by the fuzzy law with inputs of control error and its derivative. The proposed technique helps reduce overshoot and oscillation of tank level signal. However, the implementation of fuzzy law in real-time is highly computational effort and the PID controller lacks sufficient capability for more complex process systems. The issues of PID controllers are resolved with respect to applying advanced or intelligent control algorithm. The research [4] proposes a control structure that consists of the back-stepping controller and the disturbance compensation based on the super-twisting observer for the double-tank process. The research in [5] investigates model predictive control for dual-tank system. Both studies [4, 5] work with the multi-input multi-output system; the tank level converges to setpoint without static error and outstanding response when the disturbance changes arbitrarily.

p-ISSN 3093-3285
e-ISSN 3093-3315
https://doi.org/10.51316/jst.190.ssad.2026.36.2.7
Received: Sep 26, 2025; Revised: Feb 13, 2026;
Accepted: Feb 25, 2026; Online: Mar 10, 2026.

Both PID controllers and advanced control techniques are typically designed based on mathematical plant models that may not accurately represent real systems because of the incompatible and non-generalizable hypothesis and the inaccuracy of model parameters. The model-dependence problem is not significant when the reinforcement learning-based control method is utilized, one of which is the deep deterministic policy gradient (DDPG) algorithm. The research in [6] deploys the DDPG algorithm to control pH and level of a continuous stirred tank reactor simultaneously. Besides, the optimal hyperparameters of the reinforcement learning model is determined by the grid search technique. The proposed DDPG outperforms the conventional PID controller concerning the settling time and three integration criteria such as the integral absolute error (IAE), integral squared error, and integral of time-weighted absolute error (ITAE). Another version named the twin delayed deep deterministic policy gradient (TD3) algorithm is applied in the research [7] to achieve a lower overshoot and shorter settling time of the temperature in the biodiesel production. The DDPG is also investigated in [8] in the typical two input–two output process control system of which the performance is evaluated by the criteria of the overshoot, the settling time, and the steady-state error. However, above studies only compare the DDPG with traditional PID controller without other effective structures such as the anti-windup PID or adaptive PID controllers.

The dual-tank system is a complex system that exhibits nonlinearity and time delay. This system finds wide applications in industries such as chemical and power plants, where even minor deviations can lead to significant financial loss and potential accidents [9]. The study adopts the DDPG-based control strategy to manage the temperature of the dual-tank system. The performance of the proposed method is compared to various conventional PID controller structures with criteria of overshoot, settling time, and steady-state error. Beside the conventional structure of the PID, the research executes the enhanced anti-windup PID using the back-calculation method and the adaptive PID controller with the proportional term changing according to the feedback error. This ensures a fair and comprehensive comparison.

This work makes three contributions: (i) a control-oriented DDPG design for a dual tank mixing system where the state/action definitions, constraints and reward shaping are explicitly tailored for process control; (ii) an enhanced critic with a dense skip connection and ReLU activation to improve feature representation and avoid vanishing gradients; (iii) a comprehensive benchmarking against conventional PID, anti-windup PID, and adaptive-P PID. The proposed DDPG achieves faster rise, near zero overshoot, and lower steady-state error while maintaining robustness under disturbances.

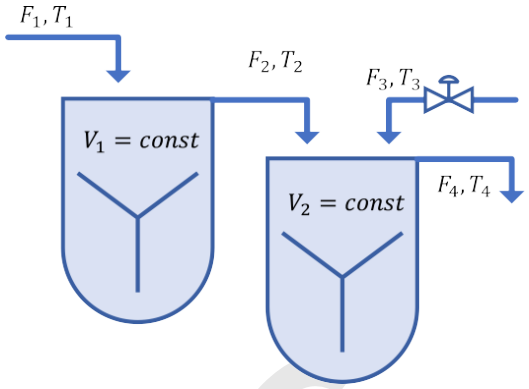


Fig. 1. Studied dual-tank mixing system configuration

The paper is presented in 5 main sections. Section 2 presents the overview of the studied dual-tank system configuration and its mathematical model. Section 3 indicates the DDPG theory and the step-by-step procedure to control the studied system. Then, Section 4 evaluates the performance of the proposed strategy by simulation and comparison to PID controllers. Finally, a few conclusions are presented in Section 5.

2. Configuration and Modeling of Dual-Tank System

2.1. System Configuration

The multi-tank mixing system is utilized widely in the industrial food or chemical processing. This study focuses on the specific case consisting of dual-tank system with overflow mechanism that illustrated in Fig. 1. Tank 1 receives an inflow and passes liquid to Tank 2, which also receives an additional controlled inflow and has an outflow. The detailed identification is listed as follows:

Flow rates:

- **F_1**: Inflow rate to Tank 1 (volume/time), typically an external input or disturbance.
- **F_2**: Outflow rate from Tank 1, which becomes an inflow to Tank 2.
- **F_3**: Additional inflow rate to Tank 2, identified as the **control variable** since it is manipulable via a valve.
- **F_4**: Outflow rate from Tank 2, exiting via the overflow mechanism.

Temperatures:

- **T_1**: Temperature of the inflow F_1 to Tank 1, an external parameter or disturbance.
- **T_2**: Temperature of the liquid in Tank 1, which is also the temperature of F_2 leaving Tank 1 (assuming a well-mixed tank).
- **T_3**: Temperature of the inflow F_3 to Tank 2, an external parameter or disturbance.

- **T_4**: Temperature of the liquid in Tank 2, which is also the temperature of F_4 leaving Tank 2 via overflow, identified as the **controlled variable**.

Volumes:

- **V_1**: Volume of liquid in Tank 1, constant due to the overflow mechanism.
- **V_2**: Volume of liquid in Tank 2, constant due to the overflow mechanism.

2.2. Construction of Studied Mathematical Model

The studied mathematical model is built based on the mass and energy balance equations for each tank. The study makes the following simplifying assumptions, typical for such systems:

- **Well-mixed tanks**: The liquid in each tank has a uniform temperature (i.e., the outlet temperature equals the tank temperature).
- **Constant volumes**: The overflow mechanism keeps V_1 and V_2 constant, simplifying mass balances to algebraic equations.
- **Constant physical properties**: Liquid density (ρ) and specific heat capacity (C_p) are constant, and there are no heat losses or external heat sources unless specified.
- **Tank 1**: Receives inflow F_1 at T_1, outflows F_2 at T_2 to Tank 2.
- **Tank 2**: Receives F_2 at T_2 from Tank 1 and F_3 at T_3, outflows F_4 at T_4 via overflow.

Based on the law of energy conservation and the assumption that the liquid volume remains constant, the temperature balance equations [5, 10] for the two tanks are derived as follows:

$$F_1 = F_2 \tag{1}$$

$$F_4 = F_1 + F_3 \tag{2}$$

$$V_1 \frac{dT_2}{dt} = F_1 (T_1 - T_2) \tag{3}$$

$$V_2 \frac{dT_4}{dt} = F_1 (T_2 - T_4) + F_3 (T_3 - T_4) \tag{4}$$

3. Deep Deterministic Policy Gradient - DDPG

3.1. General Architecture of DDPG

Deep Deterministic Policy Gradient (DDPG) is an off-policy reinforcement learning algorithm that operates online without requiring a model of the environment. It extends Deterministic Policy Gradient (DPG) to continuous action spaces while incorporating stabilization techniques from Deep Q-Network (DQN) [11]. Structurally, DDPG adopts the Actor–Critic architecture in continuous action spaces, which enables learning a deterministic policy directly. In addition, DDPG incorporates two key mechanisms inherited from DQN: (i) the use of an Experience Replay buffer, which mitigates correlations in training data by storing past experiences and sampling them randomly; and (ii) a separate target network, which stabilizes learning by decoupling the target calculation from the online network updates.

In Double DQN, the optimal action is selected by evaluating all possible actions through the Q-value function, expressed as (5).

$$\max_a Q_{\theta_Q}(s, a) = Q_{\theta_Q}\left(s, \operatorname{argmax}_a Q_{\theta_Q}(s, a)\right) \tag{5}$$

However, in continuous action spaces, evaluating every possible action becomes computationally infeasible due to the infinite number of potential values. To address this limitation, DDPG replaces the discrete action search with a deterministic policy network $\mu_{\theta_\mu}(s)$, which directly outputs the most suitable action for a given state. This actor network learns to produce actions that maximize the critic's Q-value estimation.

The critic network is updated based on the action generated by the actor, and the actor is trained to maximize the critic's output:

$$\theta_\mu \leftarrow \operatorname{argmax}_\theta Q_{\theta_Q}\left(s, \mu_{\theta_\mu}(s)\right) \tag{6}$$

By substituting the discrete action search with a deterministic policy, DDPG employs an actor–critic architecture comprising two separate neural networks: the actor network $\mu_{\theta_\mu}(s)$, responsible for generating actions conditioned on states, and the critic network $Q_{\theta_Q}(s, a)$, which evaluates the quality of these actions. This architecture is illustrated in Fig. 2 [12] and enables DDPG to operate effectively within continuous action spaces, distinctly different from the discrete action framework of Double DQN.

Specifically, the actor network deterministically maps states to actions, differing from stochastic policies where actions are sampled from a probability distribution $p(a|s)$ with $0 < p(a|s) < 1$. In DDPG, the policy is deterministic such that $p(a|s) = 1$ for the selected action, thereby reducing variance and improving stability in continuous control tasks.

To ensure that the actions produced are within valid limits, the actor's output layer typically uses bounded activation functions such as *tanh* or *sigmoid*, restricting the output within predetermined intervals (e.g. [-1, 1] for *tanh*).

The optimization of the actor network parameters leverages the chain rule to propagate gradients from the critic's Q-function with respect to the policy parameters:

$$\frac{dQ_{\theta_Q}}{d\theta_\mu} = \frac{dQ_{\theta_Q}}{d\mu} \cdot \frac{d\mu}{d\theta_\mu} \tag{7}$$

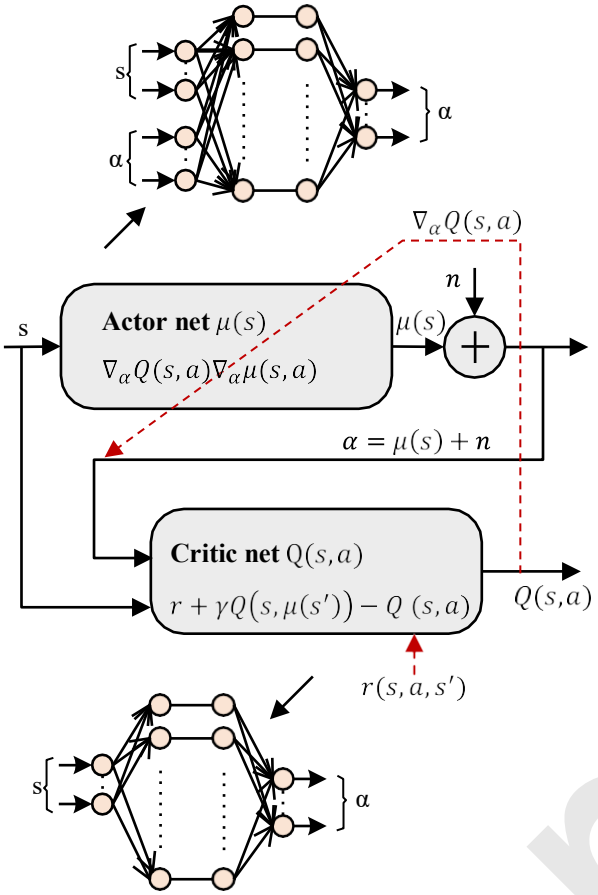


Fig. 2. DDPG algorithm structure

Here, $\frac{dQ_{\theta_Q}}{d}$ denotes the gradient of the Q-value with respect to the action, and $\frac{d\mu}{d\theta}$ is the gradient of the actor policy with respect to its parameters. This gradient propagation from critic to actor facilitates efficient policy improvement during training.

3.2. Construction and Optimization of the Loss Function

In the DDPG architecture, two separate neural networks, an Actor network and a Critic network, are trained simultaneously, each with its own objective and loss function. The Critic is responsible for estimating the action-value function $Q(s, a)$, while the Actor learns to generate optimal actions that maximize this estimated value.

To train the Critic, a loss function is constructed as the mean squared error between the estimated Q-value and a target value. For a minibatch of M experience samples (s_i, a_i, R_i, s'_i), the loss function is defined as:

$$J = \frac{1}{M}\sum_{i=1}^{M}(y_i - Q(s_i, a_i))^2 \quad (8)$$

where:

$$y_i = R_i + \gamma Q'\left(s'_i, \mu'(s'_i)\right) \quad (9)$$

is the target Q-value, computed using the target Critic and Actor networks with fixed parameters θ_μ. The discount factor $\gamma \in [0, 1]$ accounts for the importance of future rewards. The use of target networks helps to reduce training variance, thereby improving the stability of the learning process.

After updating the Critic, the Actor network is trained to produce actions that lead to higher Q-values according to the Critic's evaluation. Since the Actor does not have direct supervision, its parameters are updated using the policy gradient derived via the chain rule, which links the Critic's output to the Actor's parameters. The gradient of the Actor's objective with respect to its parameters θ_μ is computed as:

$$\nabla_{\theta_\mu} J = \frac{1}{M}\sum_{i=1}^{M} G_{\mu_i} G_{Q_i} \quad (10)$$

with:

- $G_{Q_i} = \nabla_a Q(s_i, a)$: gradient of the Critic output with respect to the action a, where a is estimated by the Actor network as $a = \mu(s)$.
- $G_{\mu_i} = \nabla_{\theta_\mu} \mu(s_i)$: gradient of the Actor output with respect to the parameters of the Actor network θ_μ.

This optimization mechanism allows the Actor network to learn a policy that improves over time according to the Critic's feedback, effectively directing the policy towards actions that maximize expected return.

In summary, the DDPG is implemented following the iterative procedure:

1) Use the Actor network to estimate the action:

$$a = \mu(s) + N$$

where N is a noise model, typically set by the user to encourage exploration.

2) Execute the action a, observe the reward R and the new state s'.
3) Store the transition (s, a, R, s') into the Experience Buffer.
4) Randomly sample a mini-batch of M transitions (s, a, R, s') from the buffer.
5) Compute the target value for the Critic network:
   - If the sampled next state s'_i is a terminal state ($s'_i = s_T$), set the target as: $y_i = R_i$.
   - Otherwise, set:
   $$y_i = R_i + \gamma Q'(s'_i, \mu'(s'_i))$$
6) Update the Critic network by minimizing the loss function over the mini-batch:

$$J = \frac{1}{M}\sum_{i=1}^{M}(y_i - Q(s_i, a_i))^2 \quad (11)$$

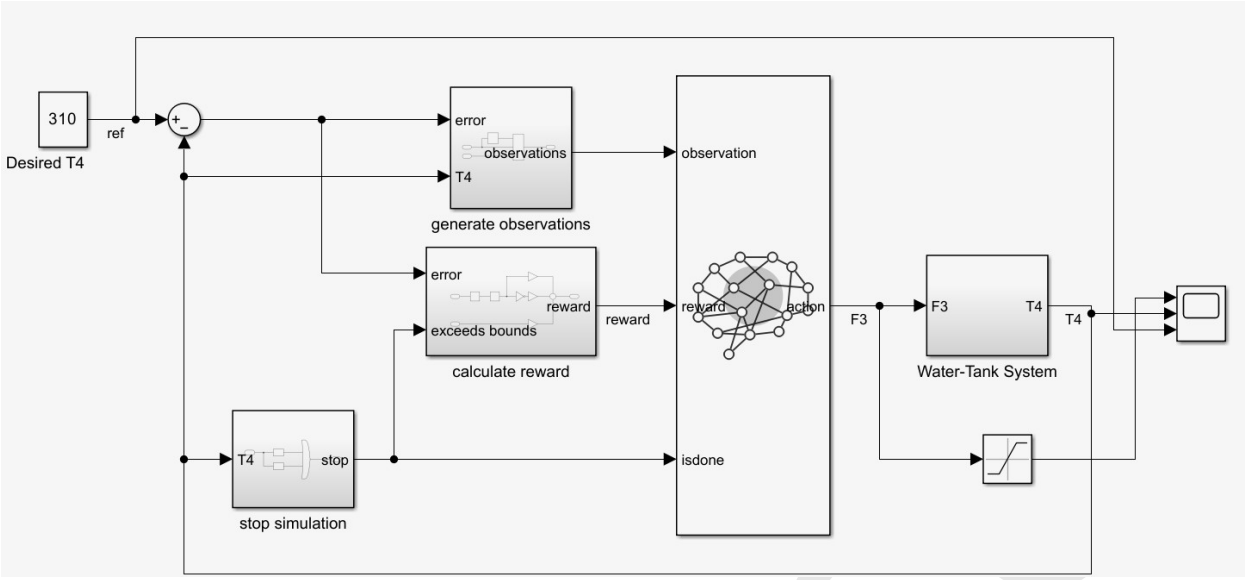


Fig. 3. Implementation of DDPG algorithm in MATLAB/Simulink

7) Update the Actor network:
   - Update parameters using the policy gradient to maximize long-term rewards.
8) Update the target networks $Q'(s, a)$ and $\mu'(S)$. Two update strategies are available:

a. Smoothing (Default):

$$\theta_{Q'} = \tau\theta_Q + (1-\tau)\theta_{Q'} \quad (12)$$

$$\theta_{\mu'} = \tau\theta_\mu + (1-\tau)\theta_{\mu'} \quad (13)$$

where τ is a smoothing factor controlling the update rate, defined by `"TargetSmoothFactor"`.

b. Periodic Update:

$$\theta_{Q'} = \theta_Q, \quad \theta_{\mu'} = \theta_\mu \quad (14)$$

3.3. Control Oriented DDPG Design for Dual Tank System

Reward shaping. Let the tracking error be defined as $e = T_4 - r$. We employ a piecewise reward to prioritize tight regulation:

$$r_t = \begin{cases} +10, & |e| < 0.2^\circ\text{C}, \\ -1, & 0.2^\circ\text{C} \le |e| < e_{\text{lim}}, \\ -100, & |e| \ge e_{\text{lim}}. \end{cases} \quad (15)$$

with e_{lim} chosen from safety limits, r is reference value of T_4.

Networks and hyperparameters. The Actor processes through two hidden layers (50, 25) with ReLU, outputting via a *tanh*. The Critic has two input branches (state 3-D, action 1-D), each with layers (50, 25), merged to a dense layer to produce $Q(s, a)$. We use Adam with learning rates $\alpha_A = 10^{-3}$, $\alpha_C = 10^{-3}$, minibatch 128, replay buffer 10^5, target update soft factor $\tau = 0.005$, and Ornstein–Uhlenbeck exploration noise.

Implementation notes. Output scaling ensures valid valve commands; input normalization stabilizes training. We pre-train under multiple set points $r \in [290, 350]K$ deploy the learned policy for evaluation.

4. Simulation Result and Discussion

4.1. Evaluation Scenario

The research focuses on the temperature management problem of the dual-tank system. The performance of the DDPG algorithm for the dual-tank system is evaluated by simulation in MATLAB/Simulink. First of all, the research constructs the mathematical model (as in Fig. 3) and implements the training process of the DDPG with the target reward of 1500. In order to obtain generally optimal control parameters, DDPG is trained with various reference temperatures in the range of (290; 350)K.

The network architecture indicated in Table 1 is designed as follows: the Critic has two inputs, a three-dimensional state and a one-dimensional action, which are processed through two hidden layers of 50 and 25 nodes before merging to output a single Q-value.
The Actor network takes the three state inputs and passes them through two hidden layers of 50 and 25 nodes, producing one action output constrained within $[-1; 1]$ by a *tanh* activation. Furthermore, the reward is selected based on the temperature error. Particularly, the reward is 10 in case of the temperature error being smaller than 0.2 and -1 in vice versa. On the other hand, the reward must receive the critical value of -100 when the error exceeds the limitation.

Table 1. Summary of Actor-Critic architecture and training hyperparameters (DDPG)

Component	Architecture (layers)	Activation	Output
Actor $\mu(s)$	Input(3) → Dense(50) → Dense(25) → Dense(1)	ReLU, ReLU, tanh	$a \in [-1, 1]$, then scaled to the actual action range
Critic $Q(s, a)$ (state branch)	Input(3) → Dense(50) → Dense(25)	ReLU, ReLU	State feature vector
Critic $Q(s, a)$ (action branch)	Input(1) → Dense(50) → Dense(25)	ReLU, ReLU	Action feature vector
Critic $Q(s, a)$ (merge)	Concatenate → Dense(50) → Dense(1)	ReLU, Linear	$Q(s, a)$ (scalar value)
Optimizer	Adam	–	–
Batch size	128	–	–
Number of episodes	350	–	–
Exploration noise	OU noise	–	–

Table 2. studied system parameters.

Parameters	Value	Unit
Uncontrolled flow rate F_1	0.5	m^3/s
Maximum controlled flow rate F_3	1.01	m^3/s
Initial temperature T_1	300	K
Initial temperature T_2	323	K
Volume of 1^{st} tank V_1	100	m^3
Volume of 2^{nd} tank V_2	75	m^3

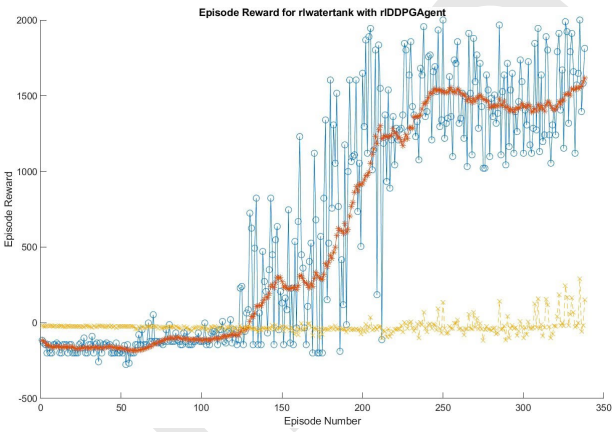


Fig. 4. Training process of agent

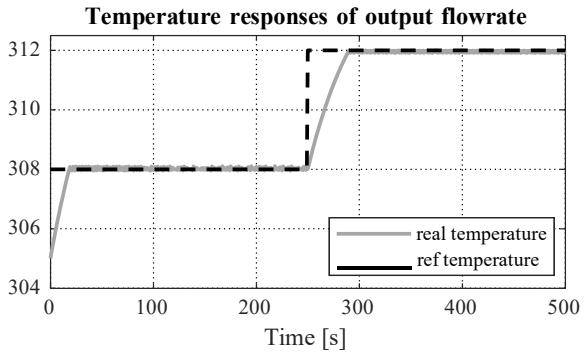
After the training process, the selected control parameter is fed into the DDPG-based temperature controller to evaluate the system performance. Then, the performance of DDPG is reinforced by comparison to various structures of the conventional PID controller such as non-anti-windup, anti-windup, and adaptive P-variable-based PID controllers considering the impactof disturbances (F_1 and T_3). The comparison is evaluated with criteria of the rising time, overshoot, settling time, and steady-state error. Model parameters are listed in Table 2.

4.2. Result of Training Procedure

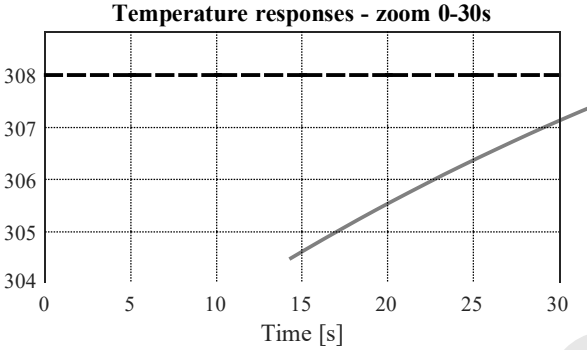
Fig. 4 displays the training process of agent implemented in MATLAB/Simulink. The blue line with circle presents the raw reward and the red line is average reward per episode. Initially, the agent exhibits low and unstable rewards, similar to a baseline performance, indicating ineffective control behavior. Around episode 120, a sharp increase in the episode reward begins, signaling that the agent has started learning an effective policy. The smoothed reward curve continues to rise and stabilizes after episode 200, with the average reward reaching above 1500, significantly outperforming the baseline. After approximately 345 iterations, the algorithm converges the expected performance. Despite some variance in episode rewards, the overall trend confirms that the agent successfully learned to optimize its control strategy over time.

4.3. DDPG Performance Discussion

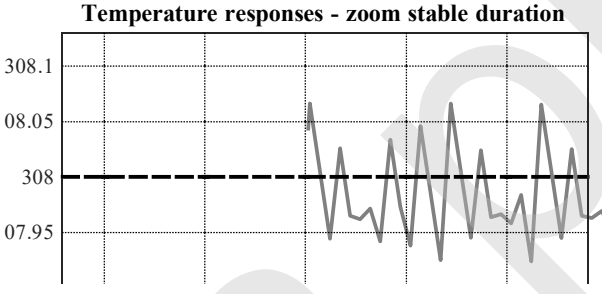
Fig. 5 illustrates the output-flow temperature response of the dual-tank system when it is controlled by a Deep Deterministic Policy Gradient (DDPG) agent after the training phase. The DDPG algorithm demonstrates good tracking of the reference value, with the controlled variable T_4 generally remaining close to the target value in the steady state without overshoot in the transient period. A similar response is observed when the setpoint changes immediately from 308 K to 312 K as illustrated in Fig. 5(a). The transient response shown in Fig. 5(b) indicates the response time of the system is approximately 18 seconds when the temperature increases from the initial value of 305 K to the first setpoint. This time is longer in case of the larger expected range of 4 K (308 K to 312 K) due to the limitation imposed on the control signal F_3. In addition, besides the convergence to the reference value, there is a slight oscillation around the setpoint in the steady state (Fig. 5c) from 307 K to 308 K. The maximum difference is only 0.1 K (or ^oC); this ripple therefore is not a cause for concern.



(a) Overall response



(b) Transient response



(c) Stable response

Fig. 5. Temperature response with DDPG algorithm

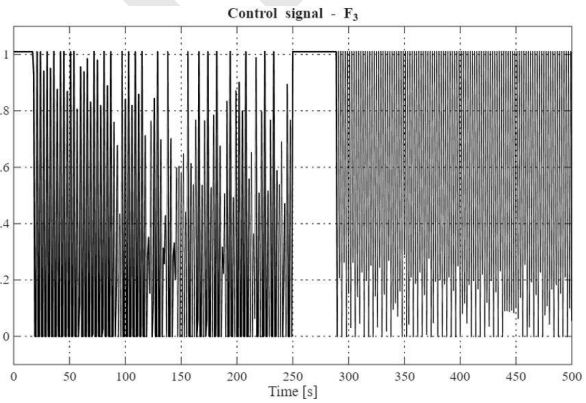


Fig. 6. Control signals F_3(t) under DDPG. The DDPG command respects valve limits

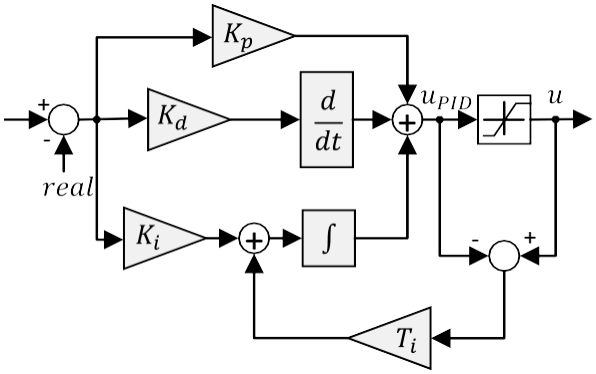


Fig. 7. Anti-windup PID controller with back-calculation method

Fig. 6 shows the control signal corresponding to the temperature response in Fig. 5. The flow F_3 varies within its specified bounds. Besides, the frequent oscillation of the control signal reveals the ripple of the temperature.

The stability analysis indicated that the system generally achieved a stable state following an initial transient period. However, persistent oscillations suggest that stability could be further improved through parameter optimization. These oscillations are likely a consequence of the exploration noise inherent in the DDPG algorithm, which, while essential for learning, introduces variability in control performance. Thus, while stability is attained in most cases, specific conditions may necessitate additional tuning to ensure uniformity.

4.4. Performance Comparison to PID Controller

4.4.1. Overview of PID structures

The performance of the proposed strategy is compared to various PID controllers. The first one is the conventional structure with fixed control parameters; the time-domain representation is given by the equation (16).

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau)\,d\tau + K_d \cdot \frac{de(t)}{d} \quad (16)$$

where: $e(t) = r(t) - y(t)$ is the error between the desired reference $r(t)$ and the actual output $y(t)$. The constants K_p, K_i, and K_d are the proportional, integral, and derivative gains, respectively.

In a PID controller, although the integral term (I) is responsible for eliminating steady-state error, this causes the 'integral windup' leading to the significant overshoot and even the instability. In order to deal with this issue, the research utilizes the back-calculation method thanks to its higher efficiency compared to the clamping technique as shown in Fig. 7. The detailed idea is to use a reset time T_i and the error between the pre-output of the PID (u_{PID}) and the saturation (u) to directly subtract the integral term. When the control signal varies in the boundary, the $u(t)$ and the $u(t)_{PID}$ have the same value;

Table 3. Comparison of control strategies

Criterion	DDPG	PID	PID + AW	PID with variable K_p	PID + AW with variable K_p
Rise time (10–90 %) [s]	≈ 55	≈ 120	≈ 80	140	75
Overshoot [%]	≈ 0	≈ 5.4	≈ 0	5.6	≈ 0*
Settling time (±0.1 °C) [s]	70	175	250	160	260
Steady-state error [°C]	≈ 0.0001	≈ 0.03	≈ 0.02	≈ 0.05	≈ 0.05

thus, the anti-windup term does not affect the controller. On the other hand, when the difference between them appears, the accumulation of the integral term is reduced. The larger the T_i, the faster the anti-windup process; however, the system dynamic can be slower. Therefore, selecting an appropriate value is critical. In this research, T_i is selected by $T_i = 0.25\frac{K_p}{K_i}$.

In addition to fixed-parameter PID controller, with respect to enhancing the performance, the adaptive PID controller can be designed with a variable proportional gain $K_p = K_o + \lambda|e$, where K_o is initial value of the proportional gain, λ is conversion coefficient, and e is the error . It means that the small error does not affect the pre-designed control parameters. Nevertheless, when the feedback error is large, the proportional term increases significantly; the control signal increases. As a result, the transition is shortened, and the system quickly moves towards stability. In addition, the anti-integral windup is also integrated to achieve more outstanding performance.

In order to compare fairly, the control parameters of the PID controller are selected by "`PIDAutotuning`" in Simulink such that the overshoot and the response time are minimized. For the studied system, K_p is 4.013738, K_i is 0.23134, and K_d is - 0.04214.

4.4.2. Comparison result

The performance comparison result between DDPG and PID controllers is shown in Fig. 8 and Table 3 concerning criteria of rising time, overshoot, settling time, and static error. It is clearly observed that the DDPG controller outperforms all PID variants in both response time and accuracy. First, it drives the liquid temperature from ≈305 K to the 310 K set-point in only 55 s (10–90 % rise time), which is roughly half the time required by the best-tuned PID with integral anti-windup and faster than any other PID controller tested. Second, the trajectory is practically critically damped: no overshoot is observed, well within sensor noise, and the system settles within a tight ±0.1 °C band in 70 s. Finally, its steady-state error is indistinguishable from zero (≈0.001 °C), demonstrating that the actor–critic network has learned to eliminate bias without relying on large integral action. In short, DDPG delivers the fastest rise, the smallest settling envelope, and essentially zero overshoot, making it the most precise and responsive strategy among the controllers evaluated.

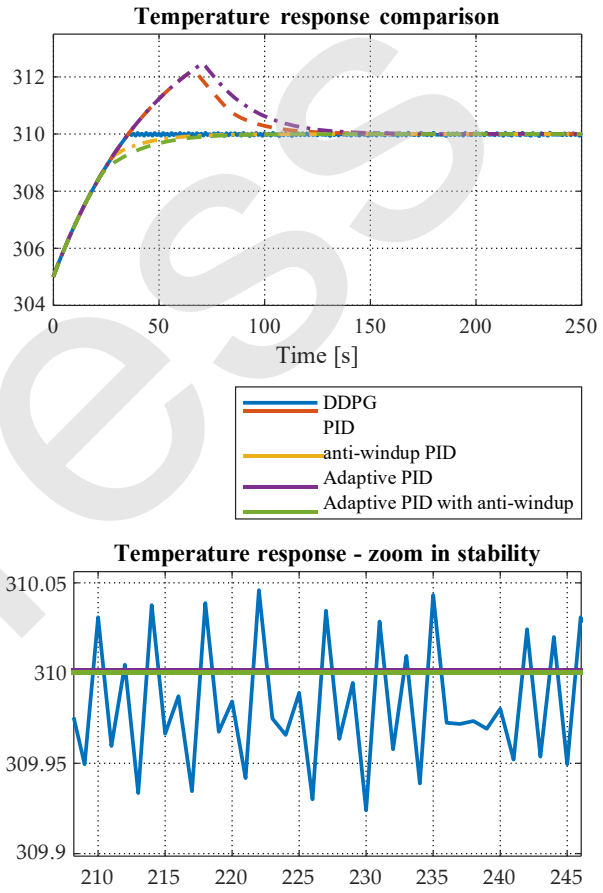


Fig. 8. Performance comparison

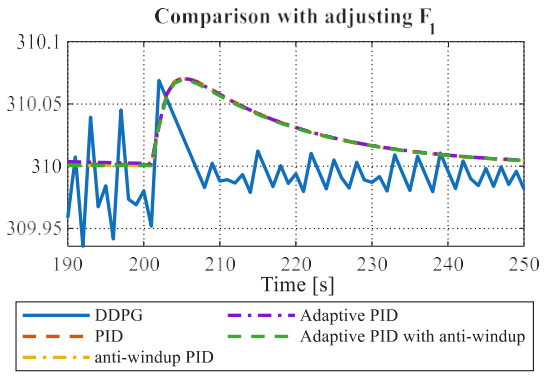


Fig. 9. Comparison result of T_4 with disturbance F_1

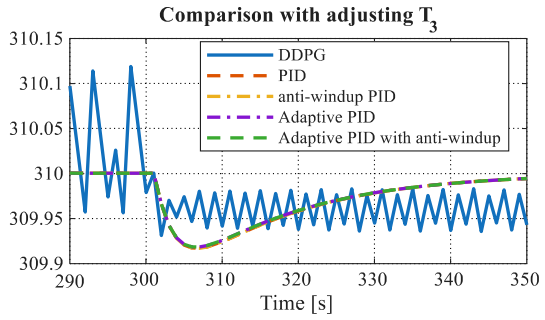


Fig. 10. Comparison result of T_4 with disturbance T_3

Fig. 9 illustrates the comparative performance under disturbance conditions, specifically when the flow rate decreases from 0.5 m^3/s to 0.1 m^3/s. It can be observed that the temperature exhibits a sudden increase, reaching an overshoot of 310.07 K, before converging to the setpoint for all control strategies. Notably, the DDPG-based controller demonstrates a faster recovery time, approximately 10s, compared to the other methods. The comparison result when changing the control flow's temperature T_3 is shown in Fig. 10, the experiment was conducted as the temperature dropped. Although the DDPG controller achieves a smaller undershoot, the system does not fully converge to the desired setpoint. This steady-state error is effectively eliminated in the PID controller due to the presence of the integral component. In general, the proposed method needs to be improved to deal with the disturbance impact completely.

5. Conclusion

Industrial control systems, particularly complex ones like the two-tank water system characterized by nonlinearity and time delay, face significant challenges from fluctuating environments and disturbances. This necessitates the application of intelligent control algorithms with autonomous learning capabilities. This study introduced a technique employing the Deep Deterministic Policy Gradient (DDPG) algorithm to manage the dual-tank system. It improved the DDPG critic network by integrating a densely connected layer and utilizing the ReLU activation function. The simulation results demonstrate that the developed strategy using DDPG significantly surpasses various PID controller structures. This superiority is evident in key metrics, including a faster convergence rate, higher setpoint tracking precision, better interference resistance, and greater overall durability. Overall, the proposed DDPG method proved capable of significantly enhancing the accuracy and stability of the dual-tank water system, highlighting its substantial potential for addressing complex industrial control problems.

References

[1] W. E. Horst and R. C. Enochs, Instrumentation and process control, Engineering and Mining Journal, vol. 181, no. 6, pp. 251–277, 1980. https://doi.org/10.1049/sqj.1959.0031

[2] E. Pivarciova and M. E. Qazizada, Applications of process control system with (SCADA) and PID controller, 12th International Conference ELEKTRO 2018, 2018 ELEKTRO Conference Proceedings, pp. 1–6, 2018. https://doi.org/10.1109/ELEKTRO.2018.8398312

[3] Y. Zhao, Y. Zhang, Y. Gong, and X. Zou, Design of double tank liquid level control system based on fuzzy PID, Journal of Physics: Conference Series, vol. 2650, no. 1, 2023. https://doi.org/10.1088/1742-6596/2650/1/012041

[4] H. Gouta, S. H. Said, and F. M'Sahli, Observer-based predictive liquid level controller for a double tank process, Proceedings of 2015 7th International Conference on Modelling, Identification and Control, ICMIC 2015, pp. 1–6, 2016. https://doi.org/10.1109/ICMIC.2015.7409371

[5] H. Gouta, S. H. Said, and F. M'Sahli, Predictive and backstepping control of double tank process: A comparative study, IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India), vol. 33, no. 2, pp. 137–147, 2016. https://doi.org/10.1080/02564602.2015.1052580

[6] C. Panjapornpon, P. Chinchalongporn, S. Bardeeniz, R. Makkayatorn, and W. Wongpunnawat, Reinforcement learning control with deep deterministic policy gradient algorithm for multivariable pH process, Processes, vol. 10, no. 12, 2022. https://doi.org/10.3390/pr10122514

[7] H. Shi, L. Zhang, D. Pan, and G. Wang, Deep reinforcement learning-based process control in biodiesel production, Processes, vol. 12, no. 12, 2024. https://doi.org/10.3390/pr12122885

[8] A. Kadu and A. Khandekar, Deep reinforcement learning-based approach for control of two input–two output process control system, International Journal on Smart Sensing and Intelligent Systems, vol. 18, no. 1, 2025. https://doi.org/10.2478/ijssis-2025-0029

[9] N. Dey, R. Mandal, and M. M. Subashini, Design and implementation of a water level controller using fuzzy logic, International Journal of Engineering and Technology (IJET), vol. 5, no. 3, pp. 2277–2285, 2013.

[10] D. E.Seborg, T. F.Edgar, and D. A.Mellichamp, Process Dynamics and Control, Second ed., Wiley, 2004.

[11] D. Silver, Deterministic policy gradient algorithms, 31st International Conference on Machine Learning, ICML 2014, vol. 1, pp. 605–619, 2014.

[12] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, Deterministic policy gradient algorithms, in Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 22–24 June 2014, pp. 387–395.