sciendo INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS

Research Article | DOI: 10.2478/ijssis-2025-0029 Issue 1 | Vol. 18 (2025)

# Deep reinforcement learning-based approach for control of Two Input–Two Output process control system

Anil Kadu* and Aniket Khandekar

Instrumentation Engineering Department, Vishwakarma Institute of Technology, Pune, India

*E-mail: anilkadu678@gmail.com

Received for publication March 01, 2025.

## Abstract

This study investigates the use of a Deep Deterministic Policy Gradient (DDPG) algorithm to control a multivariable coupled system, specifically a two input–two output (TITO) system. Traditional control methods, such as proportional–integral–derivative (PID) controllers and decoupling techniques, often face limitations in handling the complex, nonlinear dynamics and interactions within Multi Input Multi Output (MIMO) systems. The DDPG-based approach, leveraging the actor-critic architecture for continuous action spaces, enables adaptive policy learning and robust performance. Experimental results demonstrate that the DDPG controller performs significantly well compared with conventional controllers, achieving minimum integral squared error (ISE), integral absolute error (IAE), and integral time of absolute error (ITAE), indicating superior performance in minimizing deviations from target levels. These findings highlight the potential of deep reinforcement learning (DRL) for advanced multivariable control, suggesting avenues for future applications in larger and more intricate industrial systems.

### Keywords

Multivariable coupled system, Deep reinforcement learning control strategy, Deep Deterministic Policy Gradient, Two Input–Two Output Process control system, Wood–Berry distillation column model

## I. Introduction

The control of systems that are interconnected with many variables has become one of the focus areas of contemporary engineering and complex industrial processes today. It deals with systems that have more than one kind of input and output, that is, the MIMO system, In reality, these systems are conglomerated and interdependent with several variables, and sometimes even allow time delays between system inputs and outputs. Conventional control methodologies such as decentralized proportional–integral–derivative (PID) controllers and decoupling techniques do not run very well against other parameters and very complex dynamics and nonlinear behavior that are present in the systems mentioned above. The limitations of these techniques have brought about the need for developing advanced methodologies that would empower more flexible and efficient ways for control solutions.

In this situation, deep reinforcement learning (DRL) stands out as a promising method. The deep deterministic policy gradient (DDPG) algorithm is particularly effective for managing MIMO systems. Using the actor-critic setup, DDPG helps learn the best actions and value functions at the same time. This allows the controller to handle situations where actions can take any value. This capability is very useful for systems with many variables, where precise and flexible responses are crucial to maintain stability and performance when conditions change.

This study explores the application of a control strategy based on DDPG into a two input–two output (TITO) system and evaluates its performance in

comparison to traditional control methods. To design and refine the performance of the DDPG agent, the work focuses on propelling the architecture structure, the reward function, as well as the network architecture. This technique outperforms standard PID-based controllers and decoupled controllers through simulation results by minimizing overshoot while producing greater stability and lower steady-state errors. The results from this study thus not only validate the effectiveness of DRL in controlling complex coupled systems, but also promise future studies on scalable applications of reinforcement learning for circuit applications in the complex and larger MIMO systems. This work exemplifies the disruptive potential of artificial intelligence on control engineering, showing how it can inspire further developments in automation and optimization of processes. Dimensioning of a control system-DRL can be easily mapped with traditional controls principles. Whereas DRL employs an intelligent agent to interact with the environment, learn from feedback, and improve over time, traditional systems rely on predefined controllers to achieve the desired performance. To bridge these methodologies, Table 1 illustrates the correspondence between DRL components and their control system counterparts.

## II. State-of-Art Methods in Literature

Martinez-Piazuelo et al. [1] propose a multi-critic reinforcement learning methodology to enhance control of dynamical systems, particularly multi-tank water systems. The selected methodology involves an unfiltered multi-critic approach, distributing value function learning across multiple critics to simplify the learning task. The filtered multi-critic method refines this by selectively back-propagating gradients based on prior knowledge of the system dynamics. The approach demonstrates improved learning speed and sensitivity to parameter changes. Key benefits include enhanced efficiency and stability in learning, while the requirement for prior knowledge about the system may limit its model-free applicability. Wameedh et al. [2] present a decoupled control scheme utilizing improved active disturbance rejection control (IADRC) for non-linear MIMO systems. The methodology involves removing input couplings via a decoupler matrix, transforming the system into Single Input Single Output (SISO) subsystems, each controlled by IADRC. This approach was selected for its robustness and model-free characteristics, effectively estimating and rejecting disturbances in real-time. Results from numerical simulations demonstrate improved output tracking and reduced control energy compared with conventional ADRC (CADRC). Advantages include enhanced disturbance rejection and reduced chattering; however, a potential disadvantage is the complexity of implementing the decoupler matrix in real-world applications. Xu et al. [3] present a DRL approach for controlling multivariable coupled systems, utilizing the proximal policy optimization (PPO) algorithm with a tanh activation function. The methodology involves training the DRL agent using random disturbance signals and initializing set points to enhance adaptability and robustness against noise. The results indicate superior control performance compared with decentralized and decoupled methods, achieving better stability and reduced overshoot. Advantages include effective handling of complex interactions without requiring detailed models. However, the slower rise time may be a disadvantage in processes demanding rapid responses, suggesting the need for tailored reward functions to optimize performance. Ye and Jiang [4] focus on optimizing control of a double-capacity water

**Table 1: Analogy of the traditional system with DRL principles**

| DRL component | Traditional control equivalent | Description |
|---|---|---|
| Agent | Controller | Decides the actions to control the system. |
| Environment | Plant/process | The system is being controlled. |
| State | System measurements | Information about the system's current status. |
| Action | Control input | Adjustments made to influence the process. |
| Reward | Error feedback | Guides the agent to improve performance. |
| Policy | Control law | Strategy linking states to optimal actions. |

DRL, deep reinforcement learning.

tank-level system using the DDPG algorithm. It employs two approaches: DDPG pure control and DDPG adaptive compensation, enhancing feedback by integrating a PID controller. The selection of DDPG is due to its effectiveness in continuous action spaces and the ability to learn from complex environments. Experimental comparisons with traditional PID and fuzzy PID controllers demonstrate superior performance in stability and precision. While DDPG shows improved adaptability and robustness, challenges include potential computational complexity and the need for extensive training data for effective learning. Almeida et al. [5] in their survey, explore the application of fractional order control techniques to MIMO systems, emphasizing the Commande Robuste d'Ordre Non Entier (CRONE) approach and fractional PID controllers. The selection of these techniques stems from their ability to manage complex interactions within MIMO systems effectively. The analysis reveals that while many studies rely on simulations, a significant portion incorporates experimental validation, particularly using (MATLAB/Simulink Software) MATLAB/Simulink. The findings highlight improved performance in various applications, including distillation columns and robotics. However, challenges persist in achieving real-time industrial applications and ensuring stability and robustness, necessitating ongoing research to address these limitations and enhance practical implementations. A novel approach combines virtual reference feedback tuning (VRFT) and Q-learning to enhance control of MIMO vertical tank systems, as presented in Radac et al. [6]. VRFT is chosen for its ability to create initial stabilizing controllers using limited input–output data, while Q-learning refines these controllers through extensive state-action data collection. This synergy allows for improved model reference tracking performance, particularly in non-linear and constrained environments. The approach effectively handles system complexities, ensuring better decoupling of control channels. However, it necessitates careful tuning of neural network parameters and learning settings, which can complicate implementation and require substantial data for optimal performance. In Hajare et al. [7], a decentralized PID controller design for TITO processes is proposed. The approach involves using an ideal decoupler to minimize interactions between system variables, enabling the design of independent SISO controllers. Higher-order controllers are derived based on specified reference transfer functions and truncated to PID form using Maclaurin series. This technique ensures robust stability against parametric uncertainties, validated through simulations and real-time experiments on a coupled tank system. The design yields smooth control actions, enhancing actuator longevity. However, the complexity of implementation and potential challenges in tuning parameters may pose limitations in practical applications. The findings in David et al. [8] significantly enhance our project by providing a robust framework for controlling multivariable systems using DDPG. Their paper emphasizes the importance of the tanh activation function and advantage normalization, which improve learning stability in complex environments. By redesigning the reward function and controller structure, it achieves precise control, addressing the challenges of multi-loop coupling. The superior performance demonstrated through MATLAB/Simulink simulations validates the effectiveness of DDPG over traditional methods. This research equips our project with advanced techniques for optimizing control strategies, ultimately enhancing system stability and accuracy in industrial applications. Reddy et al. [9] utilize the Firefly Algorithm to optimize decentralized PID controllers for TITO systems, addressing the challenges posed by interdependent inputs and outputs. The approach involves simultaneously tuning PID parameters to minimize peak overshoot and settling time, leveraging the algorithm's efficiency in handling complex optimization problems. Comparisons with traditional direct synthesis techniques demonstrate that this approach yields improved performance metrics, such as reduced overshoot and faster settling times. While the Firefly algorithm enhances controller performance, it may require careful parameter selection and computational resources, which could impact its practicality in real-time applications. Khalid et al. [10] propose a novel approach utilizing Twin Delayed Deep Deterministic Policy Gradient (TD3) for optimizing PID controller gains in load frequency control of renewable-integrated power systems. This choice stems from TD3's ability to handle continuous control actions and overcome limitations of traditional tuning methods. The approach involves training multiple TD3 agents to minimize frequency and tie-line power deviations across interconnected areas. The findings demonstrate significant improvements in system stability and response times, outperforming conventional techniques like genetic algorithms and particle swarm optimization. However, the complexity of implementing DRL in real-world scenarios poses challenges for practical application. Mohamed and Vall [11] propose a decoupling method to be applied to obtain a decoupled system. This reduces the

interaction between the loops, making it easier to design and tune the controllers. Continued efforts are focused on improving system robustness and performance through innovative control strategies, modeling techniques, and advancements in communication technologies. In Refs [12, 13], the Non-dimensional Tuning (NDT) and Particle swarm optimization (PSO)-based decentralized PID controller approach is presented for Wood–Berry (WB) distillation process, but the decoupling process can be complex and may require significant computational effort, especially for systems with high interaction between loops. In Ref. [14], the focus is on tuning of decentralized Proportional Integral (PI) and PID controllers for TITO processes. They propose methods for optimizing these controllers to improve performance and stability in multivariable systems. Qiu et al. [15] explore the application of the DDPG algorithm within the context of energy harvesting (EH) wireless communication systems. The authors focus on optimizing resource allocation and communication strategies to improve energy efficiency and system performance. In Ref. [16], multi-agent DRL method effectively minimizes control errors in multi-area power systems, reducing load and renewable power fluctuations. In Ref. [17], the proposed DRL-based MIMO controller effectively controls non-linear energy storage in isolated microgrids with variable renewable energy sources and varying system inertia. Du et al. [18] present an innovative approach toward managing heating, ventilation, and air conditioning (HVAC) systems in residential settings using DRL. The focus is on creating an intelligent control strategy that optimizes energy efficiency while maintaining comfort across multiple zones in a residential environment. Ho et al. [19] explore the implementation of the DDPG algorithm for controlling a double inverted pendulum system mounted on a cart. The primary focus is on achieving both the swing-up and balance control tasks, which are fundamental challenges in robotics and control engineering. Lengare et al. [20] explored the limitations of traditional control strategies for MIMO processes and introduced decentralized control approaches that reduce the complexity of interdependencies between system variables. This work provides practical insight into improving the stability and efficiency of large-scale industrial systems.

## III. Proposed Methodology

This section discusses the system overview of the multivariable coupled tank system and the following subsection presents the DDPG algorithm.

### a. Multivariate coupled system

To analyze and control multivariable systems, this study leverages transfer function matrices, which have been extensively utilized for representing dynamic relationships within control systems. Figures 1 and 2 illustrate the structure and coupling mechanisms in MIMO control systems. In a MIMO control system, the system has multiple controlled variables (CVs) influenced by multiple manipulated variables (MVs). Figure 1 shows the general structure of a MIMO control system, in which $C(s)$ represents the controller and $G(s)$ represents the plant (the controlled object).

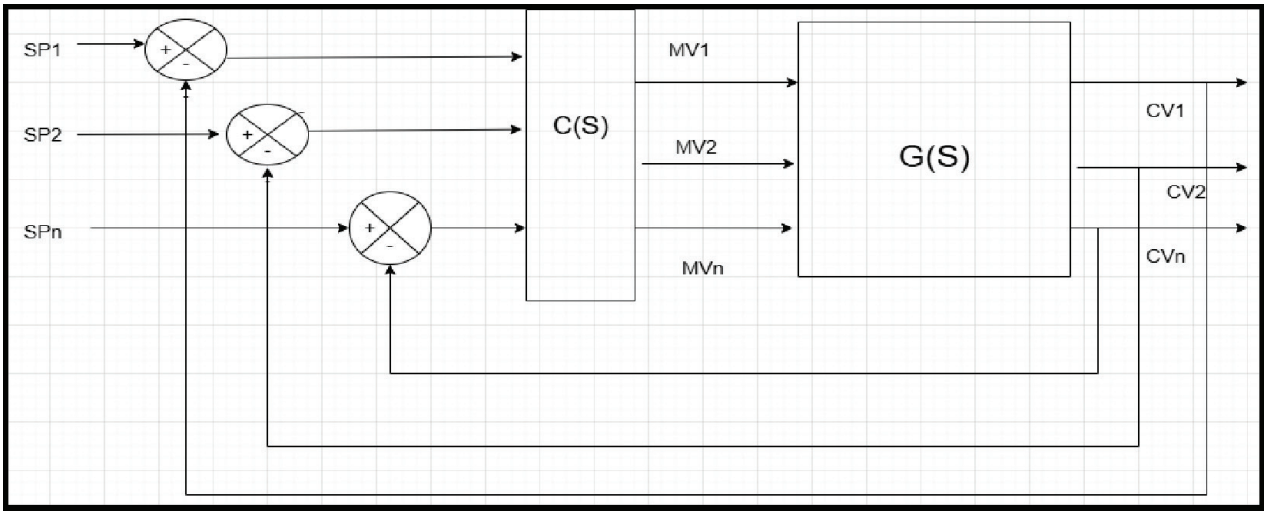


Figure 1: Overall structure of the MIMO control system.

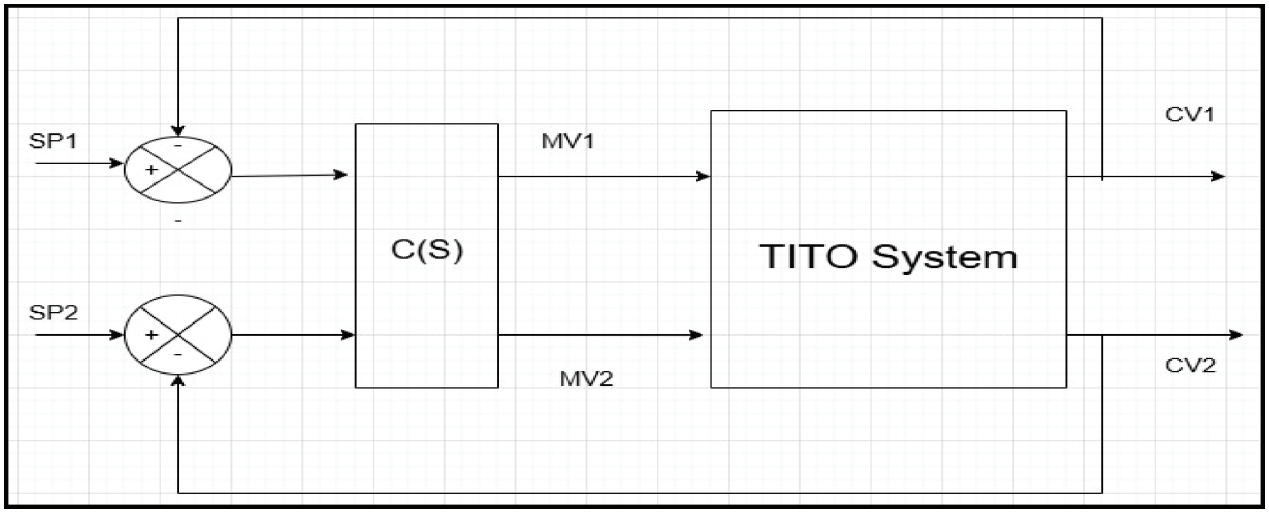


Figure 2: TITO system with controller (TITO). TITO, two input–two output.

Each setpoint has a corresponding error signal (*e*), which is fed to the controller. The controller generates the MVs that interact with the plant to adjust the CVs to the desired levels.

In multivariable control, a system is considered decoupled if each MV only affects its corresponding CV without influencing other loops. However, if there is mutual influence between loops (where an MV in one loop affects the CV in another), the system is referred to as coupled. This coupling complicates control because changes in one part of the system can impact multiple outputs. The pilot-scale distillation column model proposed by WB is represented by the following transfer function [13]. This system features an eight-tray configuration and a reboiler, designed to facilitate the separation of methanol and water. Figure 2 provides a detailed view of a TITO system. This structure serves as an example of a coupled system where each input can impact both outputs. The transfer function matrix of pilot-scale distillation column model is represented in Eq. (1):

$$G(s) = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & -\frac{18.9e^{-3s}}{21s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & -\frac{19.4e^{-3s}}{14.4s+1} \end{bmatrix} \quad (1)$$

This TITO transfer function highlights the multivariable nature of the process, featuring significant time delays and notable interactions between the input and output loops. Achieving effective chemical separation from complex mixtures remains a challenging problem for chemical engineers. Although numerous methods exist to tackle separation issues, distillation column control stands out as a widely used and cost-effective solution.

## b. DRL framework for control of TITO system

In [21] a well-known reinforcement learning algorithm that employs deep neural networks (DNN) is DDPG, a model-free actor-critic approach that has proven to be effective in a range of control issues. The DDPG algorithm serves as a foundation for the control policy for a TITO system. Based on the Actor-Critic framework, the design takes advantage of the DDPG algorithm's ability to deal with continuous action spaces to make the model well-equipped to tackle the dynamic and coupled variables that define TITO systems. The DDPG method combines two neural networks: a Critic network to assess the possible rewards of every action and an Actor network to produce optimal actions from state observations. Through a decomposition of the controller into critical components, including the Critic network, Actor network, state, action, and reward function, this section gives a descriptive overview of the DDPG-based controller as well as the key technical building blocks that contribute to its proficiency in continuous control settings.

### *b.i. Overview of DDPG algorithm*

In [22] is shown the DDPG algorithm as an off-policy, model-free reinforcement learning algorithm

developed for continuous action space environments. It is based on the deterministic policy gradient (DPG) aspects and deep Q-learning methods and is applicable for systems where the actions are continuous instead of discrete. DDPG uses two main network architectures: an Actor network, which produces a deterministic action a for a state *s*, and a Critic network, which estimates the *Q*-value function *Q* (*s*, *a*) and examines the reward potential of the state–action pair. Both networks are trained with backpropagation, whereby the policy gradients update the Actor network and Time Delay (TD) error corrects the Critic network.

One of the primary reasons for selecting DDPG over other reinforcement learning approaches is the nature of the control problem. DDPG is specifically designed for problems where the action space is continuous. The system we are working with involves a continuous action space, where actions are not discrete but instead require fine-grained adjustments. While other approaches like Q-learning or Deep Q-Networks (DQN) are effective in environments with discrete action spaces, they do not scale well to continuous actions due to the limitations of using a discrete action set. Moreover, the Actor-Critic architecture provides a balanced approach to policy learning and value estimation, improving both stability and efficiency. Another advantage of DDPG is that it uses a deterministic policy. This works well in continuous action spaces, as a deterministic policy directly outputs a specific action rather than sampling from a distribution.

The Critic network in DDPG uses a *Q*-value function inspired by Bellman's equation, which seeks to minimize the temporal difference error between predicted and actual rewards. For each time step *t*, the Critic network aims to satisfy as in Eq. (2):

$$Q(s,\ a) = E\left[\mathrm{r} + \gamma\ \mathrm{Q}(\ s',\ a')\right] \tag{2}$$

where *Q*(*s*, *a*) is the action-value function, representing the expected return after taking action *a* in state *s*; *r* is the reward received after taking action a and transitioning to state *s*′; γ is the discount factor; *a*′ is the next action taken, possibly under the policy π; and *E* denotes the expectation. The Actor network, on the other hand, optimizes its policy by directly mapping states to actions and relies on policy gradients to maximize the expected reward function *J* as seen in Eq. (3):

$$\nabla_\theta J = E_{s\sim\rho_{\pi},\, a\sim\pi_\theta}\left[\nabla_\theta Q(s,a)\right] \tag{3}$$

where $\pi_\theta$ represents the policy parameterized by $\theta$.

A key component of DDPG is the use of target networks for both the Actor and Critic networks, where target networks are soft-updated copies of the original networks, which stabilize learning by reducing correlations between target and current *Q*-values as seen in Eq. (4):

$$\theta_{\text{target}} \leftarrow \tau\theta + (1-\tau)\ \theta_{\text{target}} \tag{4}$$

where $\tau$ is a small constant, ensuring that target networks update slowly, thus preventing the instability commonly encountered in reinforcement learning with neural networks. The DDPG algorithm is designed to gradually refine the policy, allowing it to continuously adapt to dynamic changes in the environment, which is particularly useful in the TITO system with continuous control requirements. In reinforcement learning, the reward function *R* directs the agent's actions and learning by providing feedback on its performance. In some tasks, sparse rewards are used, where the agent is rewarded only when system requirements are met and otherwise penalized. However, in complex multivariable systems, sparse rewards can hinder training, so a dense reward function is applied here, for loop $y_1$. The dense reward function is formulated as in Eq. (5):

$$r_1 = \begin{cases} \alpha_1\left(-\dfrac{e_1^2}{\eta_1^2}+1\right), |e_1| \le \eta_1 \\ -\beta_1\,|e_1|\ , |e_1| > \eta_1 \end{cases} \tag{5}$$

where $\alpha_1$ and $\beta_1$ are adjustable parameters, and $\eta_1$ represents the error threshold for Loop 1. The threshold $\eta$ varies between loops based on different system error requirements. When c≠0, this reward function enables the agent to receive a nonzero reward regardless of the error's magnitude. Intuitively, this function progressively rewards the agent's performance, even when it deviates substantially from the target, which provides early-stage motivation during training. Similarly, for Loop 2, the reward function is given as in Eq. (6):

$$r_2 = \begin{cases} \alpha_2\left(-\dfrac{e_2^2}{\eta_2^2}+1\right), |e_2| \le \eta_2 \\ -\beta_2\,|e_2|\ , |e_2| > \eta_2 \end{cases} \tag{6}$$

To summarize, the reward function *R* for the multivariable coupled system depicted in Figure 2 is

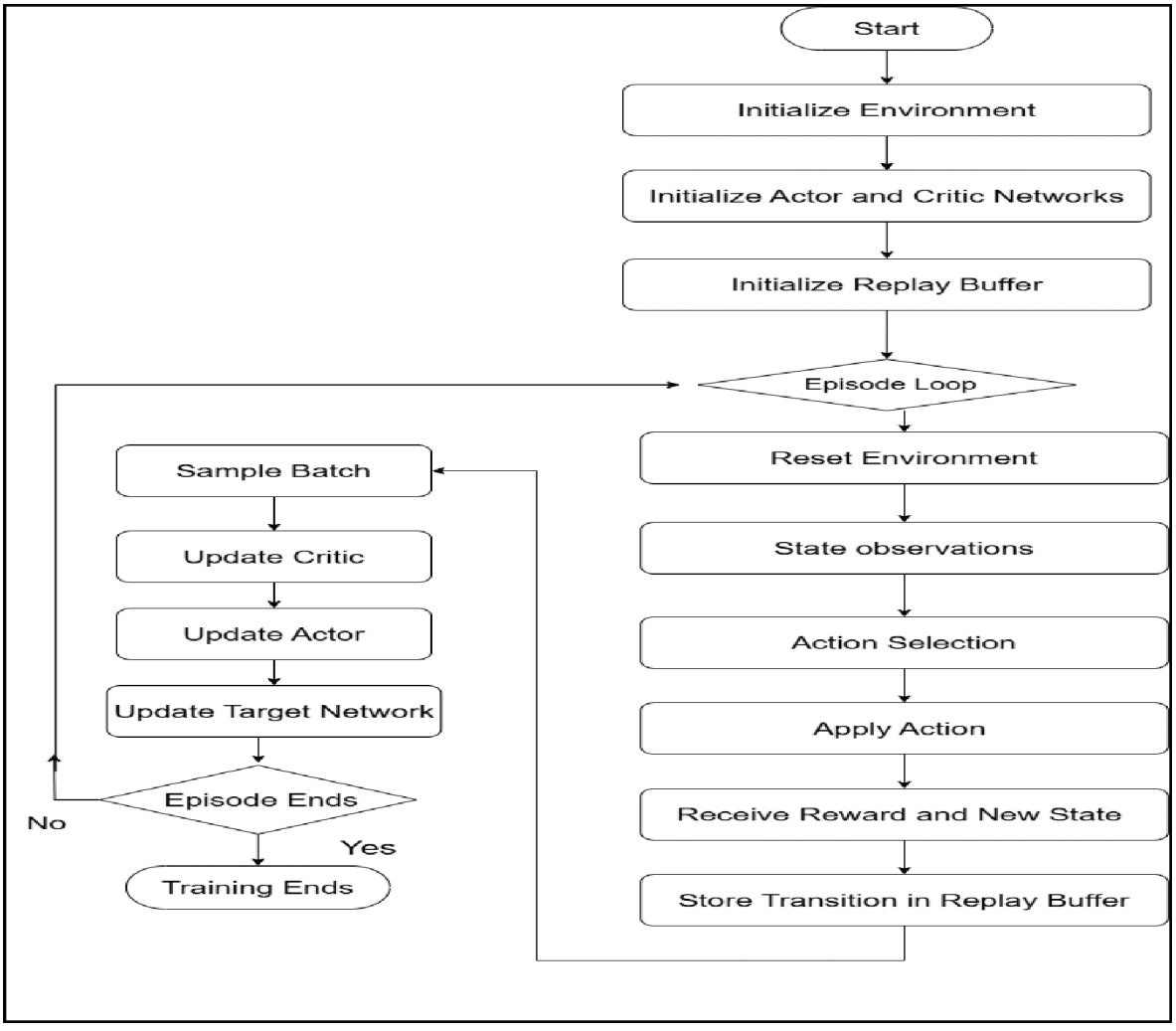


Figure 3: Simple flow chart of TITO control system using DDPG. DDPG, deep deterministic policy gradient; TITO, two input–two output.

expressed in Eq. (6). In this study, the parameters are set as follows: $\alpha_1 = 10$, $\beta_1 = 0.5$, $\eta_1 = 0.1$; $\alpha_2 = 10$, $\beta_2 = 1$, and $\eta_2 = 0.1$. The overall reward function is defined in Eq. (7):

$$R = r_1 + r_2 \tag{7}$$

## b.ii. Design of agent

*Critic network design for DDPG*

The Critic network in DDPG is responsible for estimating the *Q*-value function, which evaluates the potential cumulative reward of a specific state–action pair. In this project's TITO system, the Critic network's architecture consists of three main paths: an observation path, an action path, and a common path that merges the outputs from the other two paths. Figure 3 presents the simple flow chart of TITO control system using DDPG and Figure 4 presents the design of Critic network for DDPG.

The observation path processes the state information, *s*, encompassing variables that reflect the system's current conditions. The action path handles the control actions generated by the Actor network, ensuring distinct processing of the input action vector. The common path then combines the outputs from both paths, computes the *Q*-value, and guides policy improvement.

*Observation path*

This path processes the state information, *s*, which includes multiple variables reflecting the system's current environment and conditions. Starting with a feature input layer, the observation path inputs a state

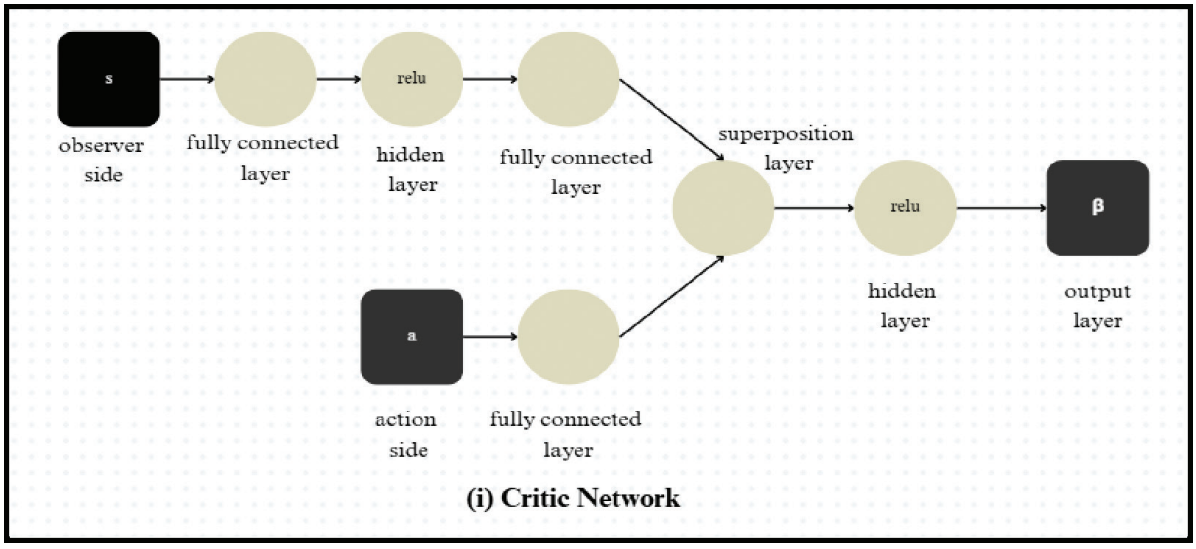


Figure 4: Critic Network design for DDPG for TITO system. DDPG, deep deterministic policy gradient; TITO, two input–two output.

vector of dimension obsInfo. Dimension (1), which includes setpoints, errors, and integral values for each control loop in the TITO system. The observation path structure is as in Eq. (8):

**obsPath** = [feature Input Layer (obsInfo. Dimension (1))

Fully connected layer (64)

Relu layer

Fully connected layer (32, Name = "obspathOutLyr")] (8)

The inclusion of ReLU activation in this path allows the network to capture non-linear state patterns effectively, and the fully connected layers with 64 and 32 neurons, respectively, enhance the path's ability to represent state features.

*Action path*

The action path separately processes the control actions generated by the Actor network, which represent the MVs in the TITO system. The structure of the action path includes a feature input layer corresponding to the action dimensions, followed by a fully connected layer of 32 neurons, which enables the transformation of action inputs into a format compatible with the common path as in Eq. (9):

**actPath** = [Feature input layer (actInfo.Dimension(1))

Fully connected layer (32, Name = " actpathOutLyr ")] (9)

Processing actions separately ensures that the network can capture distinct representations of the action space.

*Common path*

The common path combines the outputs from the observation and action paths to compute the *Q*-value for the state–action pair. Using an addition layer, the outputs of both paths are summed, which is then followed by a Relu activation to introduce non-linearity, and a final fully connected layer with a single neuron as in Eq. (10):

**commonPath** = [Addition layer (2, Name = "add")

Relu layer

Fully connected layer (1, Name = "QValve")] (10)

The resulting $Q(s, a)$ is thus an approximation of the expected reward associated with taking action $a$ in state $s$, guiding the policy improvement by updating the Critic network parameters to minimize the temporal difference error as seen in Eq. (11):

$$L(\theta) = E_{(s,a,r,s')}\left[\left(r + \gamma\, Q(s', a') - Q(s,a)\right)^2\right] \quad (11)$$

This update encourages the Critic network to predict more accurate *Q*-values, which are then used to refine the policy of the Actor network.

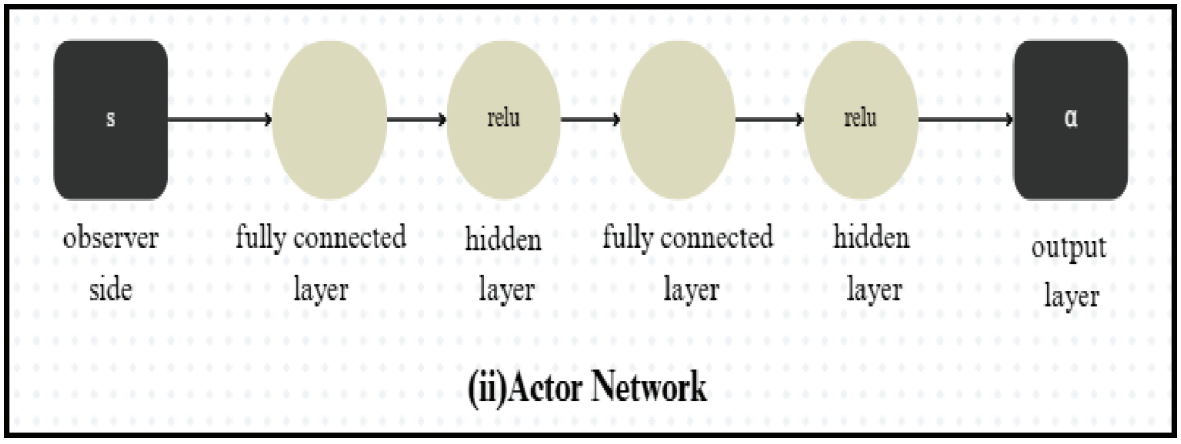


Figure 5: Actor network design for DDPG for TITO system. DDPG, deep deterministic policy gradient; TITO, two input–two output.

*Actor network*

The Actor network in DDPG generates the optimal action $a = \pi(s)$ or a given state $s$. This deterministic policy directly maps states to actions and is updated by leveraging the gradients from the Critic network. The Actor network's architecture includes a series of dense layers, where each layer contributes to processing the state information and outputting the action vector within bounded values. The structure of the Actor network is demonstrated in Figure 5.

This network begins with an input layer that aligns with the dimensions of the state vector. The hidden layer consists of three neurons, optimized for low-dimensional action spaces, followed by a tanh activation layer, which ensures that actions are constrained within the range [–1, 1], preventing excessive actions that could destabilize the system. The final layer outputs a vector with dimensions matching the action space, providing the TITO system with continuous and bounded control inputs. The Actor network updates its policy to maximize the Critic's *Q*-value, following the gradient of the objective function $J(\theta)$ as in Eq. (12):

$$\nabla_\theta J \approx E_{s\sim D}\left[\nabla_a Q(s,a \mid \theta_Q)\ \nabla_{\theta\pi}\ (s \mid \theta)\right] \tag{12}$$

This update rule ensures that the Actor network iteratively refines its policy by taking actions that maximize the expected future rewards as estimated by the Critic network.

*State and action representations*

For this TITO system, the state $s$ includes multiple input features necessary for accurate system control. These features include setpoints, current errors, and integral values associated with each loop in the TITO configuration. Formally, the state vector can be represented as in Eq. (13):

$$S_t = \left[SP_1,\ SP_2,\ e_1,\ e_2,\ \int e_1\ dt, \int e_2\ dt\right] \tag{13}$$

where Set Point (SP) is the set point, $e$ represents the current error, $\int e\,dt$ is the integral of the error. The action $a$, produced by the Actor network, adjusts the MVs within each control loop. It is represented as in Eq. (14):

$$A_t = \left[\ MV_1,\ MV_2\ \right] \tag{14}$$

These continuous actions allow the agent to produce smooth and adaptive responses, making the DDPG method suitable for TITO systems requiring precise and gradual adjustments.

*Training parameters and configurations*

A reinforcement learning agent can be trained successfully if training parameters and hyperparameters are chosen judiciously. The learning process of the agent is tracked by these parameters and the performance of the agent in optimal control affects them. For stability and convergence in the training process, critical parameters like learning rate, discount factor, and episode number must be properly set [23, 24].

Table 2 summarizes the main parameters used in configuring the DDPG agent for implementation over the TITO system.

**Table 2: Parameters for configuration of DDPG agent**

| Parameter | Description | Value |
|---|---|---|
| Discount factor (γ) | Future reward discounting | 0.99 |
| Target smooth factor (τ) | Target network update rate | 0.001 |
| Actor learning rate | Learning rate for actor updates | 0.0001 |
| Critic learning rate | Learning rate for critic updates | 0.001 |
| Mini-batch size | Sample size for experience replay | 64 |
| Experience buffer length | Total memory for experience replay | 1,000,000 |

DDPG, deep deterministic policy gradient.

### *b.iii. Learning rate and discount factor (γ)*

The learning rate determines how much the weights of the model are adjusted based on the error signal. The higher the learning rate, the quicker it can converge but potentially destabilize, whereas the lower the learning rate, the more stable it is but could weaken convergence. By trial and error and grid search, the learning rate for the Actor was chosen as 0.0001 and for the Critic as 0.001 since they performed the best in the experiment.

The discount factor determines the value of future rewards compared with current rewards. A high discount factor, like 0.99, encourages the agent to value current rewards while considering long-term consequences. Especially useful for off-policy algorithms like DDPG, where a higher discount factor ensures stable learning and decision-making, this trade-off helps the agent learn a stable and effective policy.

### *b.iv. Experience replay and soft updates (target networks)*

Through training, experience replay breaks correlations over time among consecutive data points by saving old experiences and taking random batches thereof. This reduces variance and raises stability by ensuring the model cannot learn from data that are strongly correlated. Experience replay ensures more stable learning and faster convergence through the elimination of such correlations.

In DDPG, there are target networks for both the Actor and Critic in order to stabilize training. Target networks are updated slowly with a small factor $\tau$ to avoid sudden changes in parameters. Soft updates provide smoother, more stable learning through less volatility in *Q*-values that is important to ensure stability, particularly in high-dimensional action space and continuous tasks.

### *b.v. Learning curves and episode rewards*

Learning curves usually plot the performance of the agent against time, with different phases. In initial training, the agent randomly explores, resulting in low and oscillating rewards. In the middle phase, the agent strikes a balance between exploration and exploitation, resulting in gradual improvement and stabilization of the reward curve. In the final stages, as the agent converges to an optimal policy, the learning curve levels off, meaning the agent has learned a stable and good solution.

Episode rewards monitor the agent's cumulative rewards per episode, which represent its performance in a task. The agent first makes random actions and receives low rewards. As the agent becomes more experienced and improves its policy, episode rewards increase slowly, with occasional variation because of exploration. Episode rewards eventually settle down, indicating that the agent has learned the optimal policy and is exploiting maximally.

These settings are carefully adjusted to ensure that the DDPG agent learns smoothly and effectively. This helps the agent create a reliable control policy tailored to the specific dynamics of the TITO system, making sure it understands and responds to the system's behavior accurately.

## IV. Results and Discussion

### a. Simulation of the model for experimental analysis

To test the proposed DDPG agent on the TITO system, simulation is carried out using (MATLAB/Simulink Software) Simulink software, version R2022. Figure 6 demonstrates the simulated TITO system in Simulink.

Figure 7 shows the reward function representing the critical component that guides agent learning within the DDPG algorithm. The reward function provides ongoing feedback to minimize errors while enhancing control performance. It shows how an error from the target is punished and how the smaller

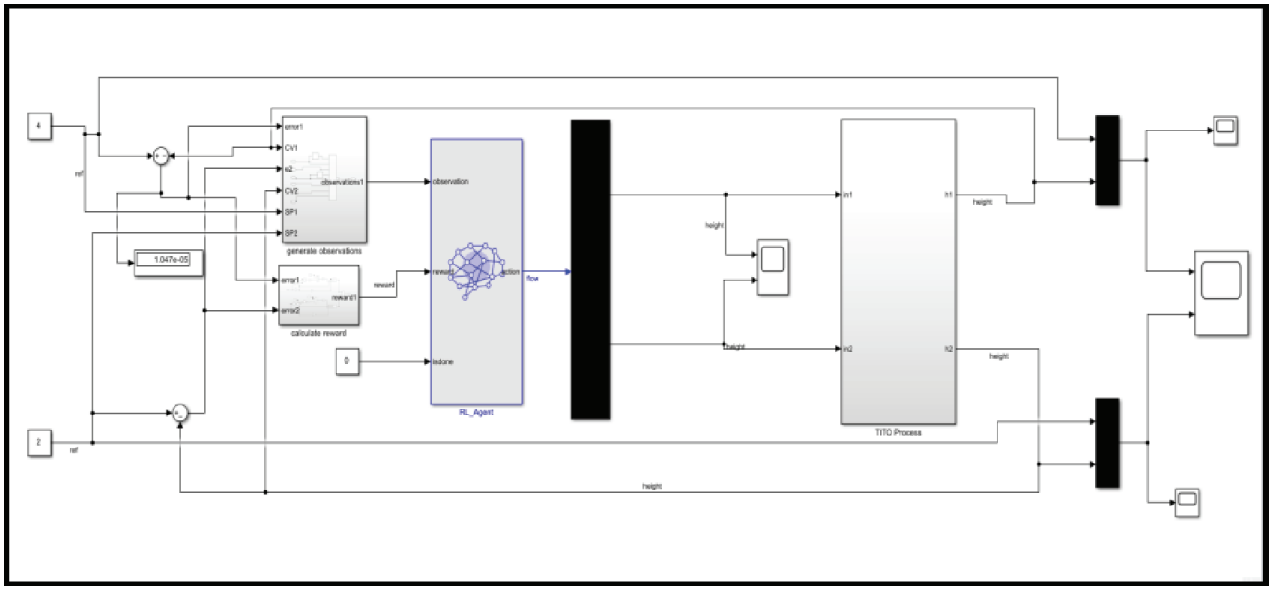


Figure 6: Simulink model for TITO system. TITO, two input–two output.

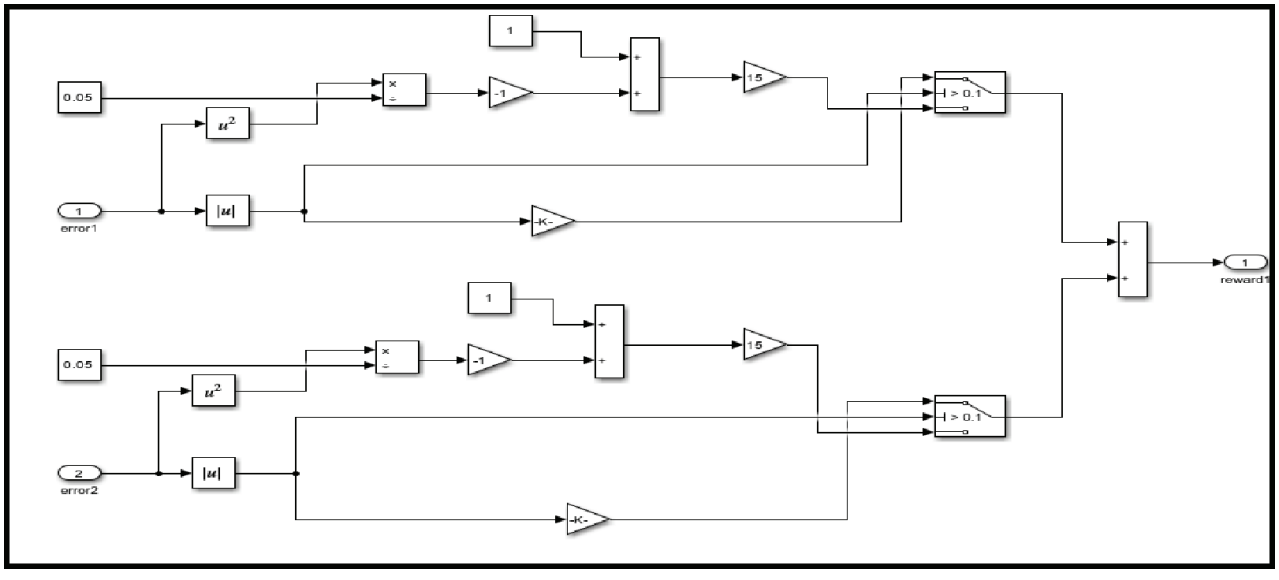


Figure 7: Reward function representation using DDPG for TITO system. DDPG, deep deterministic policy gradient; TITO, two input–two output.

the error, the more reward is achieved, which is aimed at incremental improvement in policy learning. The diagram could depict a curve illustrating the relationship between magnitude of error and associated reward value, stressing the non-linear character of the feedback mechanism.

## b. Training and validation of results using DDPG on the simulated TITO system

The training and validation of results using DDPG on the simulated TITO system is done for the set-point tracking and disturbance rejection.

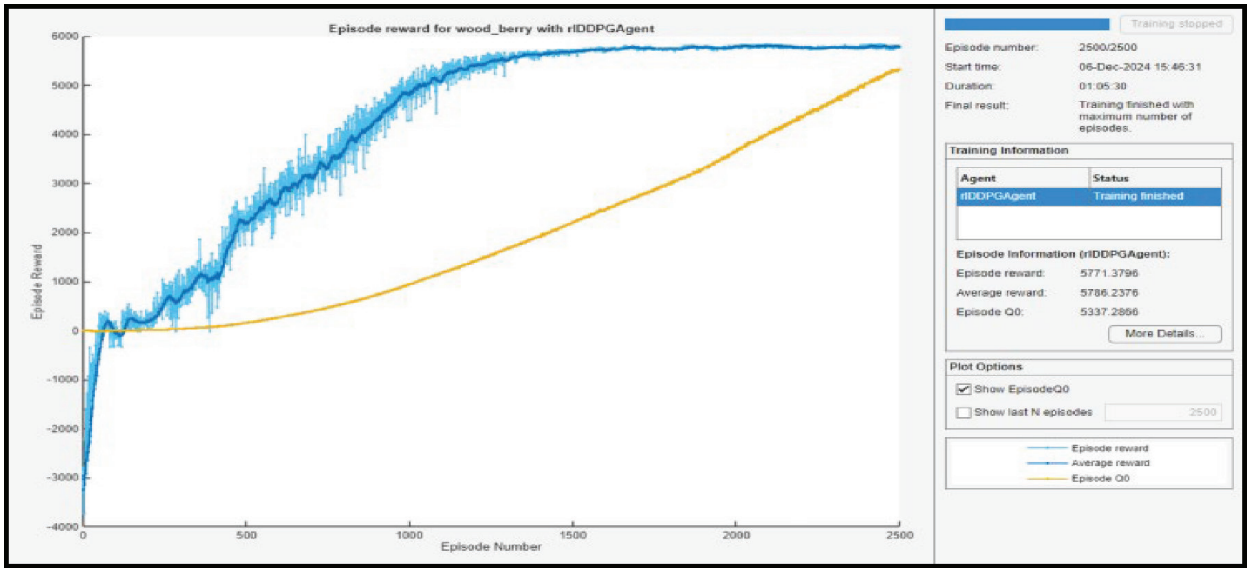


Figure 8: Training performance of the DDPG agent for TITO for set point tracking. DDPG, deep deterministic policy gradient; TITO, two input–two output.

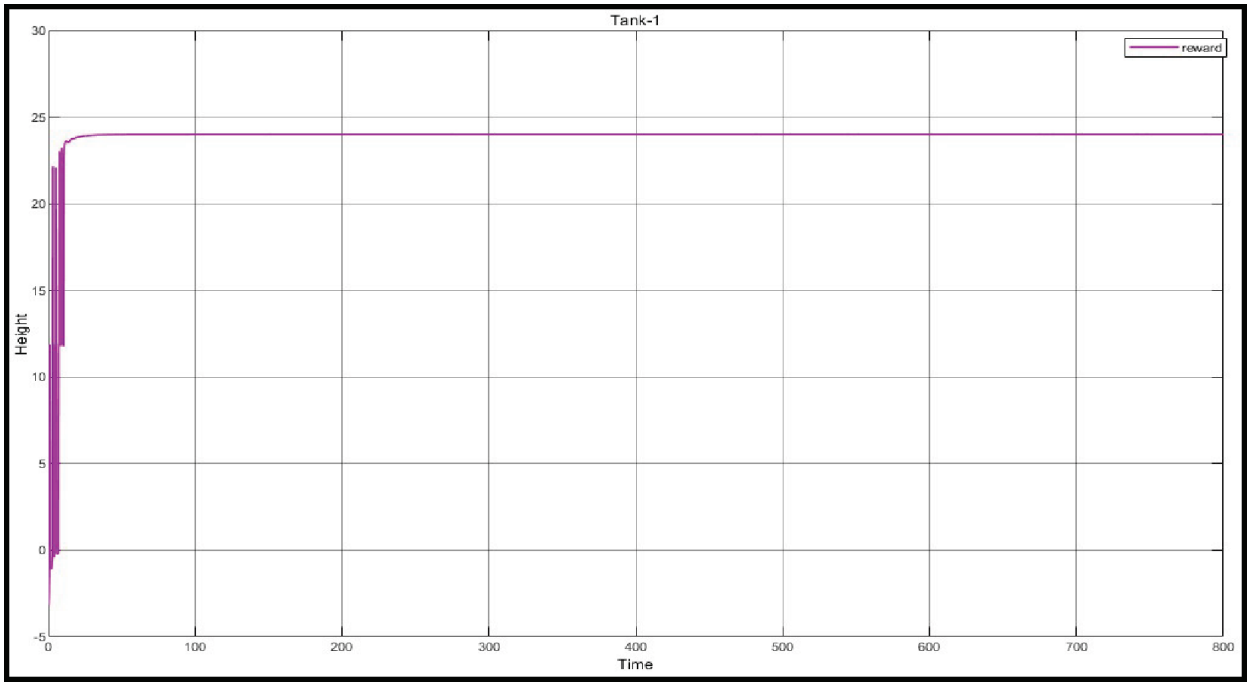


Figure 9: Reward function progression of the DDPG agent for TITO system for set point tracking. DDPG, deep deterministic policy gradient; TITO, two input–two output.

### *b.i. Training and validation results*

The training curve of the DDPG agent is shown in Figure 8, which depicts a steady improvement in the performance of the agent while it learns the dynamics of the TITO system for set point tracking. It can be seen that at around 2,500 episodes, the agent starts achieving constant and high rewards, thus indicating that the policy has converged to an optimal strategy for water level control. The graph shows an initial

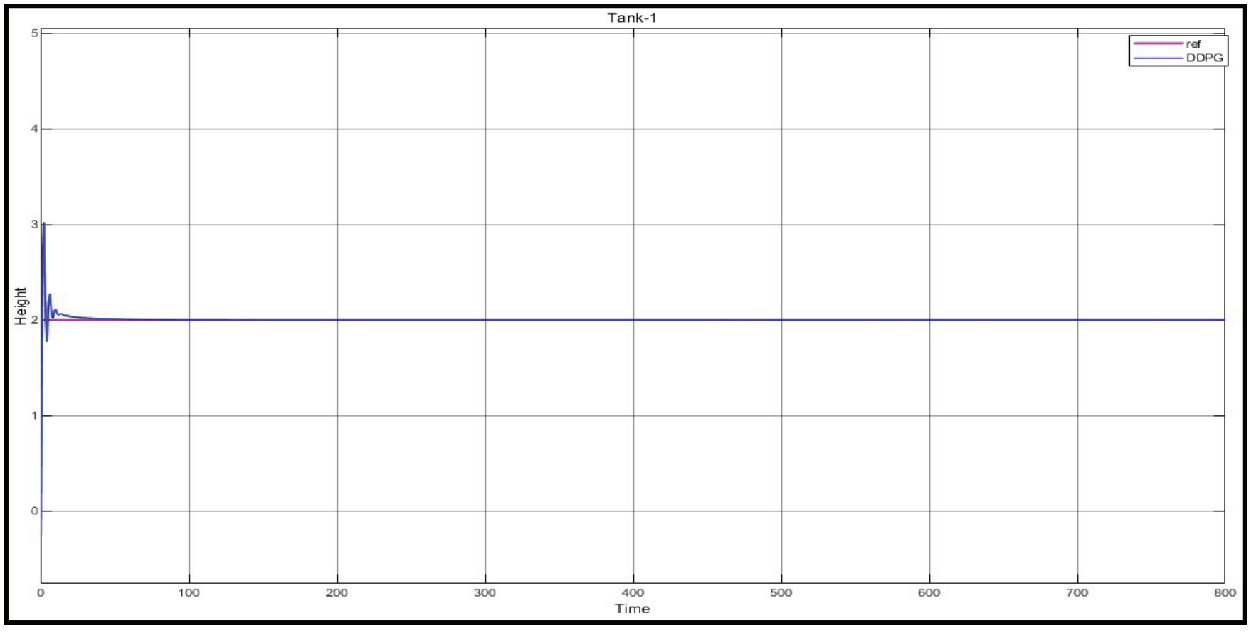


Figure 10: Simulation of transfer function on Loop 1.

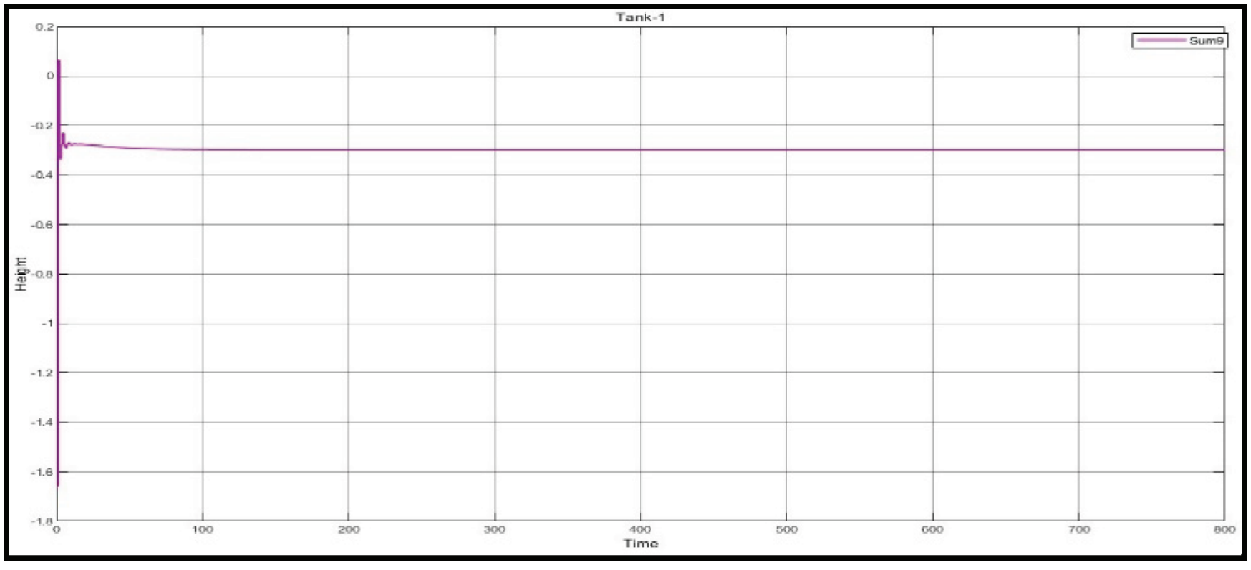


Figure 11: MV values for Loop 1. MV, manipulated variable.

unstable phase and then a stable reward curve where the agent reaches a maximum average reward of about 5,766 at the end of training.

This demonstrates that the agent learns and adapts well over time to achieve reliable control for the TITO system.

### *b.ii. Reward function behavior*

The reward function, which punishes large deviations and high control inputs, guides the agent toward efficient and accurate control. In the course of time, the agent receives higher rewards, depicting how there

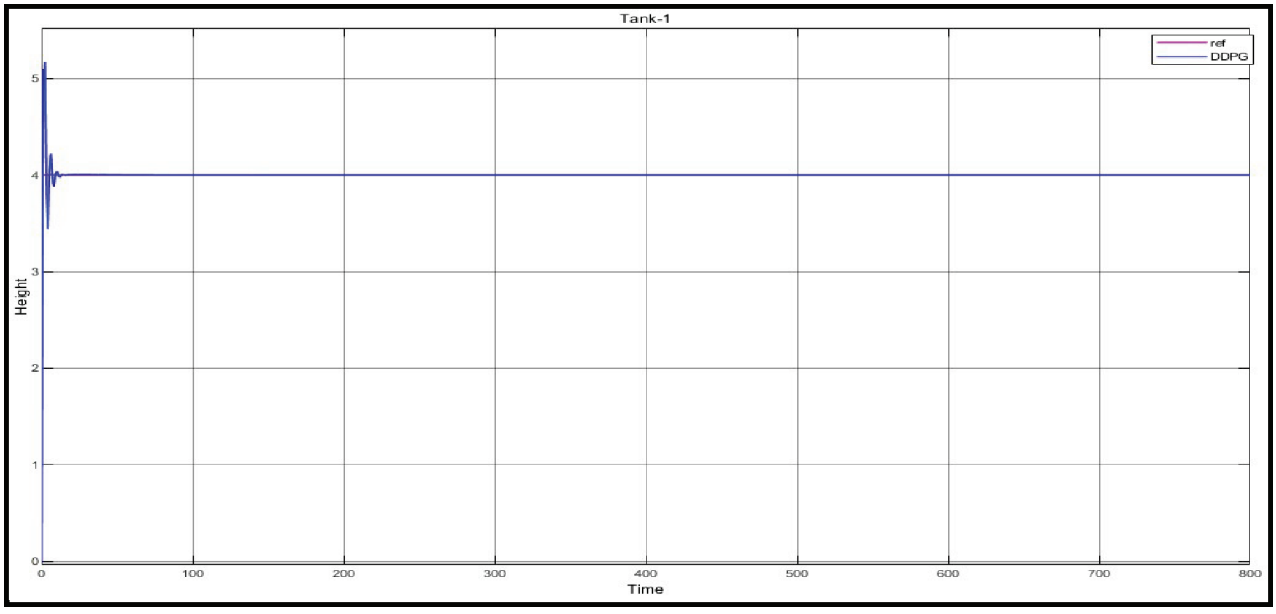


Figure 12: Simulation of transfer function on Loop 2.

is an optimization between minimizing the errors of water level and energy consumption. Figure 9 demonstrates the rewards by the reward function of the DDPG agent for TITO system for set point tracking.

The graph in Figure 9 shows the upward increase in rewards as the agent reduces its deviation. For example, the reward becomes stabilized at around +24 toward the later episodes when there is efficient control and a minimum penalty.

### *b.iii. Simulation for transfer function of the TITO system*

Simulations were conducted to analyze the transfer function responses of the TITO system under the DDPG-based control strategy. The results for Loop 1 demonstrate excellent trajectory tracking, with the system quickly achieving the desired set points while minimizing overshoot and steady-state errors. These simulations validate the controller's robustness in managing disturbances and maintaining performance in coupled dynamic systems.

Figure 10 Response of Loop 1 to set point changes, showcasing stability and rapid settling time under the DDPG control strategy.

Figure 11 shows manipulated variable (MV1) response for Loop 1 in the coupled system. The control method stabilizes MV1 after an initial transient phase, ensuring smooth and steady operation over the simulation period.

For Loop 2, the control system demonstrated similar efficiency, handling cross-coupling effects effectively. The system achieved the target set points with minimal delay and exhibited consistent performance in the face of disturbances.

Figure 12 illustrates the controlled response of Loop 2, highlighting its stability and precise tracking performance under dynamic conditions.

Figure 13 illustrates the manipulated variable (MV2) response for Loop 2 in the coupled system. The DRL controller mitigates oscillations and stabilizes MV2, demonstrating effective control of the multivariable system dynamics.

## c. Comparison of proposed methods with other traditional methods

To evaluate the effectiveness of the proposed DRL-based control strategy, we compared it against several traditional and contemporary control methods, including NDT [14], Mvall [11], and Wang et al [20].

$$K_{\text{NDT [PI]}} = \begin{bmatrix} 0.41 + \dfrac{0.074}{s} & 0 \\ 0 & -0.12 - \dfrac{0.024}{s} \end{bmatrix}$$

$$K_{\text{Mvall [PI]}} = \begin{bmatrix} 0.1532 + \dfrac{0.0067}{s} & 0 \\ 0 & -0.0266 - \dfrac{0.0015}{s} \end{bmatrix}$$

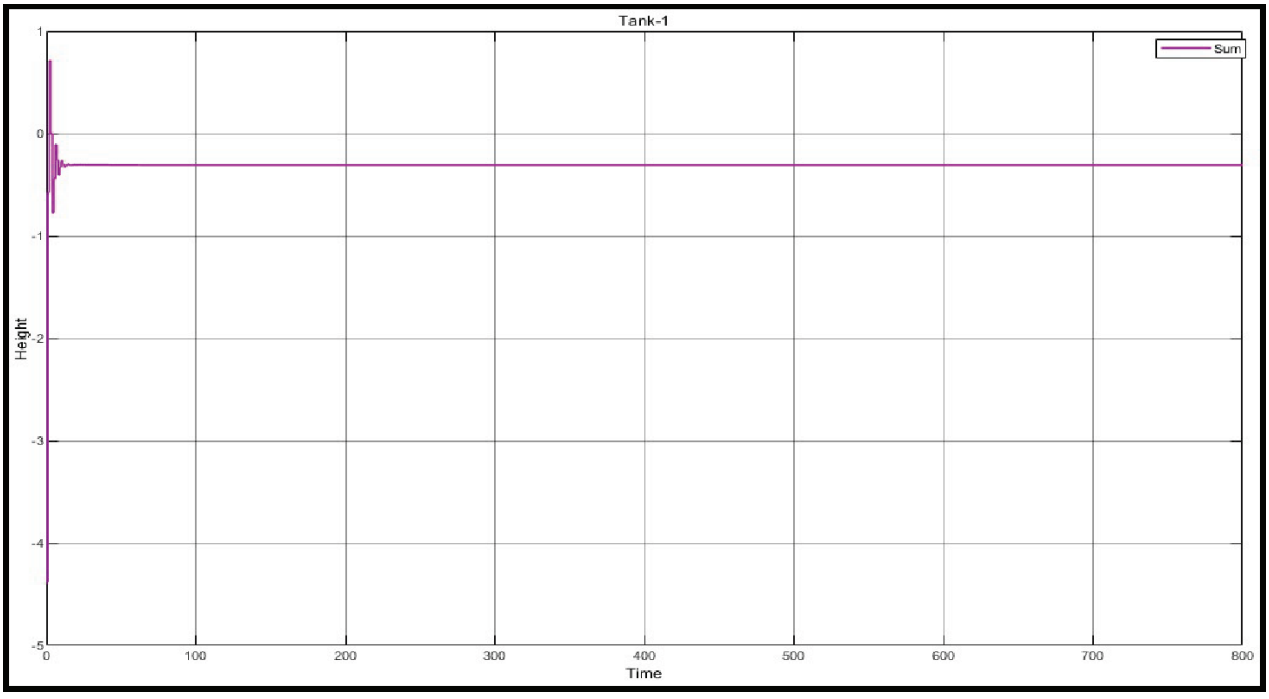


Figure 13: MV values for Loop 2.

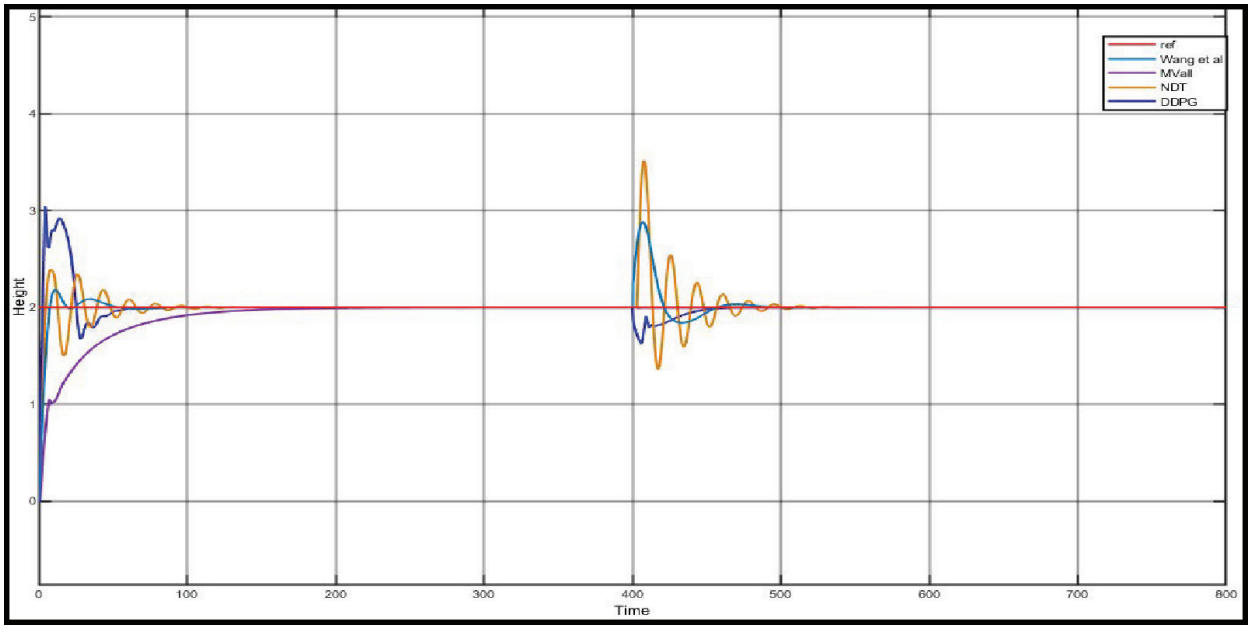


Figure 14: Comparison of proposed method on Loop 1 with traditional methods. DDPG, deep deterministic policy gradient.

$$K_{\text{wang [PID]}} = \begin{bmatrix} 0.216 + \dfrac{0.076}{s} + 0.017 & 0 \\ 0 & -0.068 - \dfrac{0.019}{s} - 0.06 \end{bmatrix}$$

The performance was analyzed on a TITO system, as shown in Figures 14 and 15.

In Loop1, the DRL-based control method demonstrated superior performance across multiple metrics. The DRL controller achieved a settling time of approximately 48 s, compared with other methods.

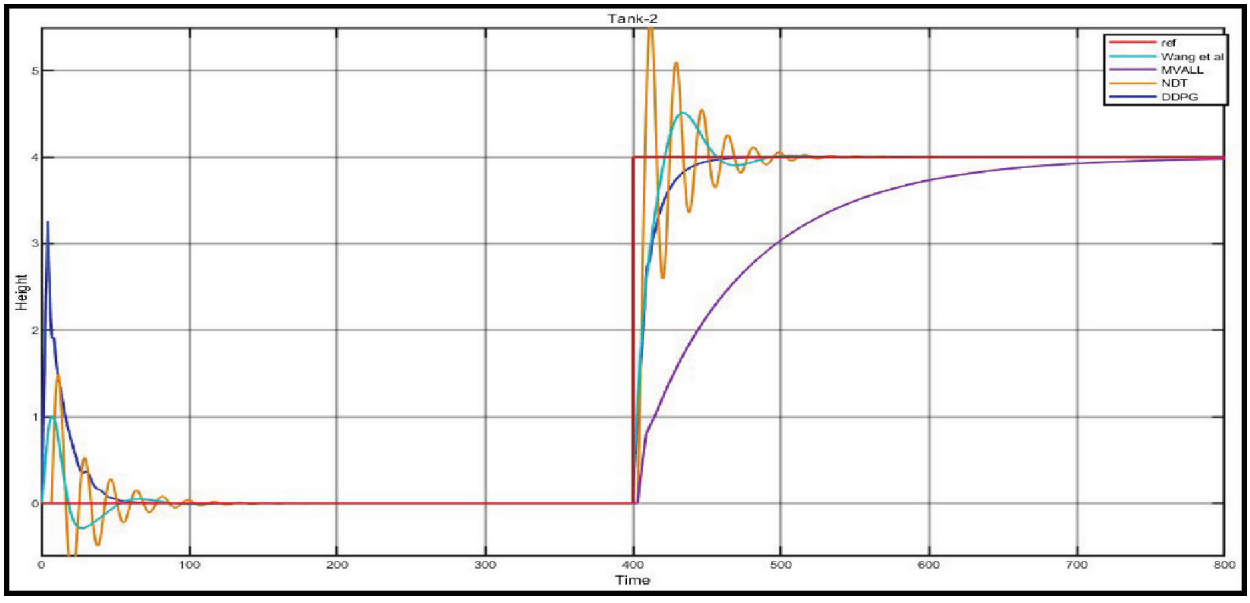


Figure 15: Comparison of proposed methods on Loop 2 with traditional methods. DDPG, deep deterministic policy gradient.

## Table 3: Performance indices of Loop 1

| Method | ISE | IAE | ITSE | ITAE | Overshoot (%) | Settling time | Steady-state error |
|---|---|---|---|---|---|---|---|
| DDPG | 18.31 | 29.92 | 722.9 | 3325 | 35 | 48 | 0 |
| NDT [PI] | 26.82 | 39.9 | 6631 | 1.032e + 04 | 25 | 100 | 0 |
| Mvall [PI] | 34.61 | 47.25 | 488.3 | 1880 | 0 | 150 | 0 |
| Wang et al [PID] | 16.26 | 24.82 | 3206 | 6517 | 20 | 53 | 0 |

DDPG, deep deterministic policy gradient; IAE, integral absolute error; ISE, integral squared error; ITAE, integral time of absolute error; ITSE, integral time squared error; PID, proportional–integral–derivative.

The DRL controller exhibits superior performance in Loop 2 when benchmarked against alternative methods. The interactions managed by DRL resulted in control dynamics that became both smoother and more precise. The DRL-based control strategy demonstrated superior performance across all evaluated metrics such as settling time, overshoot, and steady-state error. The examination of these results demonstrates DRL's proficiency in handling multivariable coupled system complexities while delivering substantial benefits compared with traditional and benchmark control methods.

### *c.i. Error metrics analysis*

The DDPG algorithm was tested against common control methods such as PI, PID, and decoupled strategies. To assess how well each method performed, metrics like integral squared error (ISE), integral absolute error (IAE), integral time squared error (ITSE), and integral time absolute error (ITAE) were used. These metrics were calculated for both loops in the system. From Figure 14 and Table 3, it can be seen that the proposed controller has less ISE, IAE, ITSE, and ITAE error and less settling time as compared with other controllers, but slightly more overshoot. The DDPG method consistently performed better than the others. It showed smaller error values and better stability for the system, especially when dealing with the complex dynamics of the TITO system.

Table 3 presents the performance indices for Loop 1, comparing the DDPG approach with traditional control methods. The DDPG controller achieved

## Table 4: Performance indices of Loop 2

| Method | ISE | IAE | ITSE | ITAE | Overshoot (%) | Settling time | Steady-state error |
|---|---|---|---|---|---|---|---|
| DDPG | 137.7 | 79.13 | 3.217e + 04 | 1.707e + 04 | 0 | 42 | 0 |
| NDT[PI] | 122.1 | 82.69 | 4.434e + 04 | 2.515e + 04 | 60 | 110 | 0 |
| Mvall [PI] | 510.3 | 275.3 | 2.228e + 05 | 1.305e + 04 | 0 | 380 | 0 |
| Wang et al [PID] | 81.82 | 61.27 | 2.947e + 04 | 1.856e + 04 | 30 | 85 | 0 |

DDPG, deep deterministic policy gradient; IAE, integral absolute error; ISE, integral squared error; ITAE, integral time of absolute error; ITSE, integral time squared error; PID, proportional–integral–derivative.

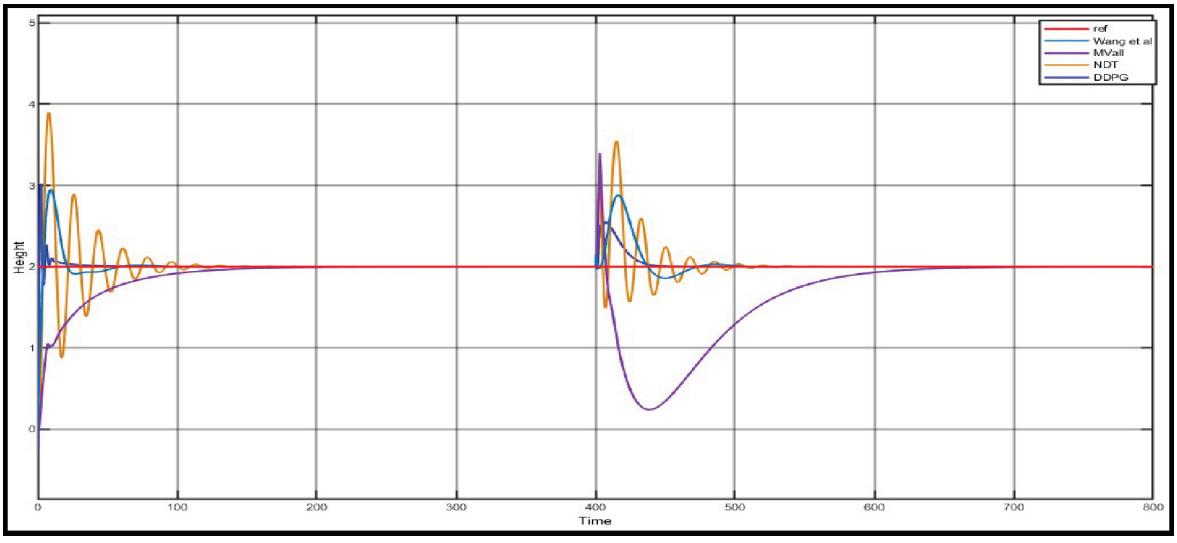


Figure 16: Response to disturbance on Loop 1. DDPG, deep deterministic policy gradient.

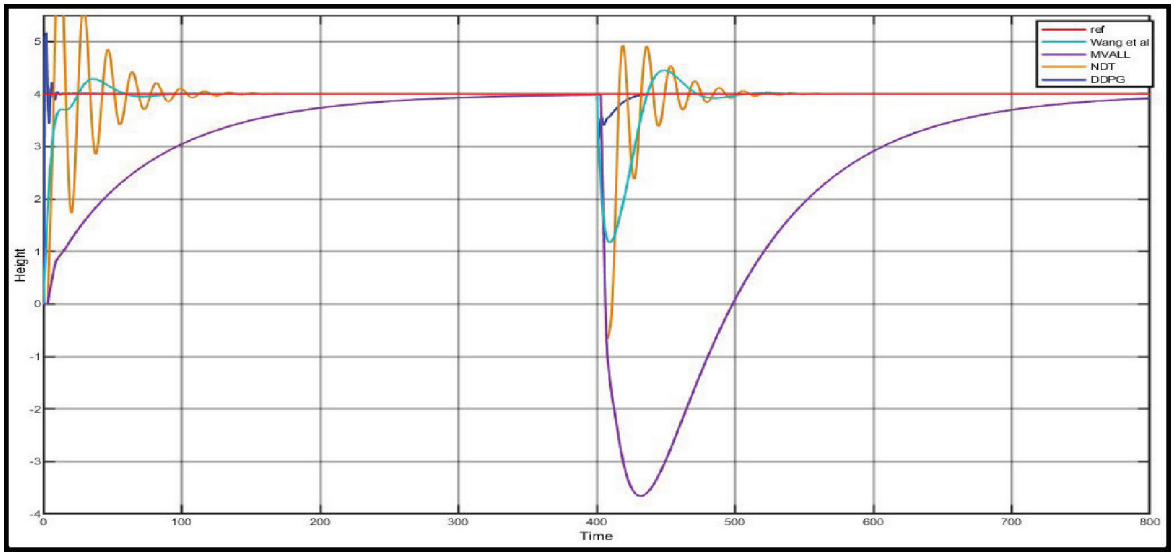


Figure 17: Response to disturbance on Loop 2. DDPG, deep deterministic policy gradient.

the lowest values across all indices, indicating superior precision and adaptability.

Similarly, Table 4 compares performance indices for Loop 2. The DDPG method's robustness is evident from its improved performance over conventional techniques, From Figure 15 and Table 4, it can be seen that the proposed controller has less overshoot and settling time as compared with other controllers.

### *c.ii. Simulation with disturbance*

To evaluate the robustness of the proposed control strategy under dynamic environmental changes, we performed simulations introducing disturbances in the multivariable coupled system. The results for Loop 1 and Loop 2 under disturbance scenarios are presented in Figures 16 and 17, respectively. In Figure 15, the height variations of Loop 1 under different control methods are illustrated. The DDPG (DDPG) controller stabilizes within approximately 48 s, exhibiting an overshoot of only 5%, compared with 15% and 10% overshoots observed for the Wang [PID], NDT[PI], and Mvall [PI] methods, respectively. These results highlight the adaptability of the DDPG approach in mitigating disturbances effectively and ensuring system stability. Figure 16 illustrates the height variations for Loop 2 under similar disturbance conditions. The DDPG controller achieves stabilization within 40 s with an overshoot of approximately 8%, closely aligning with the reference trajectory. By contrast, methods like those of Wang [PID], NDT[PI], and Mvall [PI] demonstrate overshoots of up to 20% and take over 85, 100, and 200 s to settle, respectively. The DDPG-based approach quickly suppresses disturbances and ensures the system returns to equilibrium efficiently, showcasing its robustness and efficiency.

## V. Conclusions

In this paper, the DRL-based approach for control of the TITO process control system is proposed. A well-known reinforcement learning algorithm that employs DNN is DDPG, a model-free actor-critic approach that has proven to be effective in a range of control issues. The DRL approach, which is outlined in this study for control of the TITO systems, relies on the algorithm for DDPG as one method because of its flexible way of handling continuous action spaces and learning stable control policies. To validate the performance of the proposed controller, the WB example is simulated using MATLAB. and the performance of the proposed controller is compared with the traditional PI/PID controller with set point tracking and disturbance rejection. Experimental results show that the proposed controller works well as compared with another controller. The novelty in this study is invoking the DDPG's model-free Reinforcement Learning (RL) to address problems affecting conventional tuning methods that rely on accurate knowledge of system dynamics and do not adapt to time-varying conditions. In that regard, the proposed method automates the tuning process, minimizing the human effort while enabling scalability when applied to industrial control systems. Among the key contributions of the study are ensuring training stability through experience replay and soft target updates, and performance feedbacks in terms of ISE, IAE, ITSE, and ITAE are also improved. In future research directions, it may be interesting to further develop this approach in conjunction with classical control or alternative algorithms that would improve computational efficiency.

## References

[1] Martinez-Piazuelo, Juan, Daniel E. Ochoa, Nicanor Quijano and Luis Felipe Giraldo. "A Multi-Critic Reinforcement Learning Method: An Application to Multi-Tank Water Systems." IEEE Access 8 (2020): 173227-173238.

[2] Wameedh Riyadh Abdul-Adheem, Ibraheem Kasim Ibraheem, Decoupled control scheme for output tracking of a general industrial nonlinear MIMO system using improved active disturbance rejection scheme, Alexandria Engineering Journal, Volume 58, Issue 4,2019, Pages 1145-1156, ISSN 1110-0168

[3] Xu, Jin, Han Li, and Qingxin Zhang. 2023. "Multivariable Coupled System Control Method Based on Deep Reinforcement Learning" Sensors 23, no. 21: 8679.

[4] Ye L, Jiang P. Optimization control of the double-capacity water tank-level system using the deep deterministic policy gradient algorithm. Engineering Reports. 2023; 5(11): e12668.

[5] Almeida, Alexandre Marques de, Marcelo Kaminski Lenzi, and Ervin Kaminski Lenzi. 2020. "A Survey of Fractional Order Calculus Applications of Multiple-Input, Multiple-Output (MIMO) Process Control" Fractal and Fractional 4, no. 2: 22.

[6] Radac, Mircea-Bogdan, Radu-Emil Precup and Raul-Cristian Roman. "Data-driven model reference control of MIMO vertical tank systems with model-free VRFT and Q-Learning." ISA transactions 73 (2018): 227-238.

[7] Hajare, V.D., Patre, B.M., Khandekar, A.A. et al. Decentralized PID controller design for TITO processes

with experimental validation. Int. J. Dynam. Control 5, 583–595 (2017).

[8] Silver David, Guy Lever, Nicolas Manfred Otto Heess, Thomas Degris, Daan Wierstra and Martin A. Riedmiller. "Deterministic Policy Gradient Algorithms." International Conference on Machine Learning (2014).

[9] M. D. L. Reddy, P. K. Padhy and I. Ahmad Ansari, "Auto-tuning Method for decentralized PID controller of TITO systems using firefly algorithm," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 2019, pp. 683-688.

[10] Khalid, Junaid & Anwari, Makbul & Khan, Muhammad & Hidayat, T. (2022). Efficient Load Frequency Control of Renewable Integrated Power System: A Twin Delayed DDPG-Based Deep Reinforcement Learning Approach. IEEE Access. 10. 1-1.

[11] Ould Mohamed and Mohamed Vall, Design of Decoupled PI Controllers for Two-Input Two-Output Networked Control Systems with Intrinsic and Network-Induced Time Delays. Acta Mechanica et Automatica, 15, 201-208, December 2021.

[12] Reshma N. Pawar and Sharad P. Jadhav Design of NDT and PSO based Decentralized PID Controller for Wood-Berry Distillation Column IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI-2017).

[13] Wood RK, Berry MW (1973) Terminal composition control of a binary distillation column. Chem Eng Sci 28(9):1707–1717.

[14] Tavakoli S, Griffin I, Fleming PJ (2006) Tuning of decentralized PI(PID) controllers for TITO processes. Control Eng Pract 14(9):1069–1080.

[15] Qiu C, Hu Y, Chen Y, Zeng B. Deep deterministic policy gradient (DDPG)-based energy harvesting wireless communications. IEEE Internet Things J. 2019;6(5):8577-8588.

[16] Yan, Z., & Xu, Y. (2020). A Multi-Agent Deep Reinforcement Learning Method for Cooperative Load Frequency Control of a Multi-Area Power System. *IEEE Transactions on Power Systems*, 35, 4599-4608.

[17] Skiparev, V., Belikov, J., Petlenkov, E., & Levron, Y. (2022). Reinforcement Learning based MIMO Controller for Virtual Inertia Control in Isolated Microgrids. 2022 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 1-5.

[18] Du, Y., Zandi, H., Kotevska, O., Kurte, K., Munk, J., Amasyali, K., Mckee, E., & Li, F. (2021). Intelligent multi-zone residential HVAC control strategy based on deep reinforcement learning. *Applied Energy*, 281, 116117.

[19] Ho, T, Tat, T, Ngo, H., Nguyen, T, Bui, D., Le, T., Le, V, & Huynh, L. (2023). Applying DDPG Algorithm to Swing-Up and Balance Control for a Double Inverted Pendulum on a Cart. Robotica & Management.

[20] Wang, Q-G., Zou, B., Lee, TH., Bi, Q. (1997). Auto-tuning of multivariable PID controllers from decentralized relay feedback. *Automatica*, 33(3):319–30.

[21] A. Alharin, T.-N. Doan, and M. Sartipi, "Reinforcement Learning Interpretation Methods: A Survey," IEEE Access, vol. 8, pp. 171058 171077, 2020.

[22] T. M. Luu and C. D. Yoo, "Hindsight Goal Ranking on Replay Buffer for Sparse Reward Environment,"IEEE Access, vol. 9, pp. 51996 52007, Mar. 2021.

[23] Kiran, M. and Ozyildirim, M. Hyperparameter Tuning for Deep Reinforcement Learning Applications. arXiv: 2201.11182. (2022).

[24] Felten, F, Gareev, D, Talbi, E.G. and Danoy, G. Hyperparameter Optimization for Multi-Objective Reinforcement Learning. arXiv: 2310.16487.(2023).