

Article

Enhancing Virtual Inertia Control in Microgrid Clusters: A Novel Frequency Response Model Based on Deep Reinforcement Learning

Adrián Criollo ¹, Dario Benavides ^{1,2,*}, Paul Arévalo-Cordero ¹, Danny Ochoa-Correa ¹,
Luis I. Minchala-Avila ¹ and Marcos Tostado-Véliz ²

¹ Department of Electrical Engineering, Electronics, and Telecommunications (DEET), Universidad de Cuenca, Cuenca 010101, Ecuador; adrian.criollo@ucuenca.edu.ec (A.C.); paul.arevaloc@ucuenca.edu.ec (P.A.-C.); danny.ochoac@ucuenca.edu.ec (D.O.-C.); ismael.minchala@ucuenca.edu.ec (L.I.M.-A.)

² Department of Electrical Engineering, University of Jaén, 23700 Linares, Spain; mtostado@ujaen.es

* Correspondence: djbenavi@ujaen.es or javier.benavidesp@ucuenca.edu.ec

Abstract

Traditional control strategies, such as droop-frequency and PI controllers, often show limited adaptability when the system operates under highly variable renewable generation and load conditions. In order to overcome this limitation, this study proposes the design and simulation of a clustered microgrid supported by reinforcement learning (RL)-based virtual inertia control. Three continuous-control RL algorithms were evaluated: Deep Deterministic Policy Gradient (DDPG), Twin-Delayed Deep Deterministic Policy Gradient (TD3), and Soft Actor-Critic (SAC). The SAC agent provided the most robust training performance, reaching stable convergence after approximately 400 episodes and a final reward close to 86 units after 600 episodes. DDPG presented the second-best behavior, whereas TD3 achieved the lowest final reward, approximately 43 units. The proposed Agent SAC-Reinforcement Learning control was tested on a two-area microgrid cluster, which demonstrated greater frequency stability. Results indicate a frequency nadir of 59.82 Hz and ROCOF of 0.1484 Hz/s, with 6.78% nadir deviation improvement and 37.23% ROCOF reduction compared to PID-based strategies.

Keywords: virtual-inertia; microgrids-cluster; dynamic estimation; stability, reinforcement learning

1. Introduction

The increasing penetration of renewable energy sources (RES) has significantly reduced the inherent mechanical inertia traditionally provided by synchronous generators, thereby posing critical challenges to frequency stability in clustered microgrid systems. Conventional droop-based and PI control strategies exhibit limited adaptability under fast RES-induced power fluctuations, which motivates the development of advanced virtual inertia control frameworks capable of coordinating interconnected microgrids, mitigating frequency deviations, and enhancing operational resilience. In addition to conventional droop and PI-based control strategies, active disturbance rejection control (ADRC) has emerged as an effective alternative for engineering systems subjected to internal uncertainties and external disturbances. ADRC estimates and compensates lumped disturbances in real time, which makes it suitable for nonlinear and uncertain dynamic systems. Moreover,



Academic Editor: Roberto Zivieri

Received: 4 May 2026

Revised: 30 May 2026

Accepted: 24 June 2026

Published: 3 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

advanced smart-material-based control systems, including piezoelectric structures and flexible beam control applications, have shown significant potential for vibration suppression, disturbance compensation, and dynamic response improvement [1].

Microgrids facilitate the coordinated integration of distributed energy resources, storage systems, and controllable loads, operating either grid-connected or islanded to improve reliability and energy efficiency [2]. Nevertheless, high penetration of wind and solar generation reduces both inertia and damping, exacerbating frequency instability under renewable intermittency and load variations [3,4]. These limitations highlight the need for adaptive, data-driven control strategies—particularly reinforcement learning (RL)—to achieve real-time and resilient frequency regulation in renewable-based microgrids.

Reinforcement learning provides an adaptive alternative for power system control by learning optimal policies directly from interaction data, without requiring explicit system models or complete future information [5]. Its data-driven structure is well suited for nonlinear renewable-based systems, enabling controllers to respond effectively to uncertainty, parameter variations, and unforeseen disturbances. Recent advances, ranging from Q-learning to deep RL architectures, demonstrate its potential for sequential decision-making in frequency regulation and energy management across modern grids [6].

The previous work [7] focused on the development of a frequency response model for virtual inertia and virtual droop control in an isolated microgrid supported by energy storage systems. In contrast, the present manuscript extends that foundation by incorporating a deep reinforcement-learning-based control layer for the adaptive tuning of virtual inertia and damping coefficients. Specifically, the proposed study evaluates DDPG, TD3, and SAC as alternative continuous-control reinforcement learning agents within the same virtual inertia control framework.

Deep reinforcement learning (DRL) extends RL by integrating neural networks to handle high-dimensional state spaces, thereby solving complex power system control problems [8]. Among DRL methods, DDPG has been widely applied to continuous control tasks through actor–critic networks, making it suitable for governor control and energy storage dispatch [9]. Similarly, SAC enhances robustness by incorporating entropy-based exploration, supporting more generalizable control policies under uncertain operating conditions [10]. Multi-agent deep reinforcement learning (MARL) further enables coordinated control among distributed agents. Frameworks such as MADDPG support centralized training with decentralized execution, leveraging local information to achieve scalable coordination [11]. This structure is particularly suitable for distributed microgrids, where communication bandwidth is limited and local autonomy is essential. Within hierarchical microgrid control, primary, secondary, and tertiary layers operate at different time scales to address fast disturbance response, nominal frequency restoration, and optimal energy dispatch, respectively [12].

In this context, advanced virtual inertia control in microgrid clusters emerges as a key strategy for improving frequency stability under high renewable penetration. While droop and PI controllers remain constrained in adaptability to fast disturbances, DRL and MARL approaches enable intelligent controllers to learn optimal policies, adapt to dynamic operating conditions, and coordinate distributed responses. Consequently, MARL-based virtual inertia control represents a strategic contribution toward reliable, sustainable, and decarbonized power systems.

1.1. Literature Review

Single-agent deep reinforcement learning (DRL) has demonstrated clear advantages over conventional frequency control methods in renewable-based power systems. In particular, TD3 reduces overestimation bias through delayed policy updates and target policy

smoothing, achieving frequency deviations within $\pm 1\%$ during transient events [13]. Similarly, DDPG has been applied to load frequency control, ensuring physically feasible policies and validated performance in multi-area systems [14]. Moreover, data-driven DDPG approaches have proven effective for emergency frequency control without requiring detailed system models, outperforming conventional schemes in diverse scenarios [15]. These findings collectively highlight the adaptability of actor–critic methods, which combine value-based stability with policy-based convergence properties [16].

Beyond single-agent approaches, Proximal Policy Optimization (PPO) and its multi-agent variants have shown strong potential for complex systems with multiple objectives. For instance, hierarchical PPO schemes reduced frequency fluctuations by 37.8% while supporting higher renewable penetration [17]. This suggests that PPO offers a balanced trade-off between convergence speed and generalization, making it suitable for practical deployment [18]. Nevertheless, convergence patterns differ: SAC converges more slowly but ensures stronger steady-state robustness, whereas DDPG converges faster but may be less resilient under severe disturbances.

Extending to multi-agent reinforcement learning (MARL), Yang et al. [19] introduced distributed architectures where each DER agent learns local policies while coordinated training optimizes global objectives. Furthermore, Rudai et al. [20] emphasized centralized training with decentralized execution (CTDE) as essential for scalability, reducing communication requirements. Recent advances, such as transfer learning and meta-reinforcement learning, accelerate convergence and improve robustness in real microgrid applications [21]. Additionally, evolutionary and attention-based MARL frameworks enhance coordination, robustness, and scalability in large-scale distributed systems [22–24].

Regarding virtual inertia control (VIC), reinforcement learning has been identified as a key enabler for frequency stability under high renewable penetration. For example, RL-based adaptive VIC allows inertia constants to vary dynamically with operating conditions, outperforming fixed-parameter designs [25,26]. Complementary approaches, such as MPC combined with RL, improve virtual inertia provision from energy storage systems [27], while derivative-based VIC simultaneously emulates inertia and damping effects [28]. Notably, extensions to DC microgrids and integration with advanced storage technologies such as SMES further broaden the applicability of VIC strategies [29,30].

Finally, reinforcement learning has also been integrated into energy management systems (EMS), where DRL and MARL frameworks optimize resource dispatch, reduce costs, and improve renewable utilization [31–33]. In summary, the literature confirms that DRL and MARL approaches not only enhance frequency regulation through adaptive virtual inertia control but also support broader objectives of resilience, cost reduction, and sustainability in modern microgrids.

1.2. Problem Research and Contribution

In real-world applications, most power systems are interconnected via a tie-line or transmission line to help each other in regulating and exchanging interchange power as illustrated in Figure 1, improving system stability, reliability, and resiliency. The trend of measured frequency in each region is an important index of the power unbalance in the interconnected area (not in the single area).

Most control schemes applied to the emulation of multiple virtual inertia in power systems rely on classical PID controllers. However, it is well known that PID control requires significant computational effort, since tuning its parameters (proportional, integral, and derivative) is not straightforward and often fails to achieve a global optimum in dynamic and highly variable environments. This limitation becomes even more critical in

interconnected microgrid clusters, where subsystem interactions increase the complexity of parameter adjustment.

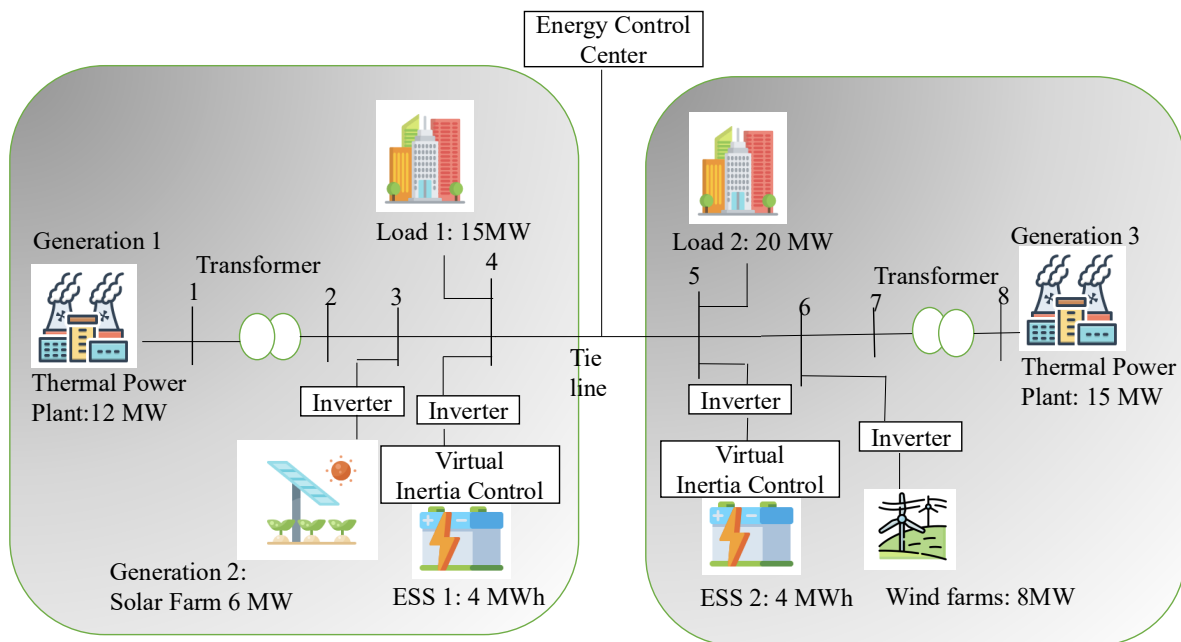


Figure 1. Energy control center based on single-agent DRL-based control framework.

The approach proposed in this study aims to overcome these restrictions by replacing traditional PID control with a reinforcement learning framework, specifically through the DDPG algorithm. This method enables an intelligent agent to autonomously learn optimal control policies, adapt to changing operating conditions, and reduce the need for exhaustive manual tuning. In doing so, it introduces a more flexible and scalable framework for virtual inertia control, fully aligned with the latest trends in advanced algorithms for optimal energy system management.

The novelty of this research lies in the development of an adaptive reinforcement learning-based virtual inertia control strategy for clustered microgrids with high renewable energy penetration. Unlike conventional PID-based virtual inertia controllers, the proposed framework enables the online adjustment of virtual inertia and damping coefficients through trained RL agents. In addition, this study provides a comparative assessment of three continuous-control algorithms, DDPG, TD3, and SAC, under the same virtual inertia control framework. The proposed approach is validated through a realistic microgrid case study based on the University of Cuenca laboratory system, considering photovoltaic generation, wind generation, diesel generation, battery energy storage, and programmable load disturbances. This allows evaluating the impact of the proposed control strategy on ROCOF reduction, frequency nadir improvement, and overall frequency stability.

The main contributions can be summarized as follows:

1. An adaptive reinforcement-learning-based virtual inertia control framework is proposed for interconnected microgrid clusters, enabling online tuning of virtual inertia and damping coefficients through DDPG, TD3, and SAC agents.
2. A two-area frequency response model is developed considering tie-line power exchange, photovoltaic and wind disturbances, diesel generation dynamics, battery energy storage, and sudden load variations.

3. The proposed method is compared with conventional PID-based virtual inertia control, demonstrating improved frequency nadir, reduced ROCOF, smoother transient response, and better adaptability under high renewable penetration.

The structure of this paper is organized as follows: Section 2 covers RL fundamentals and the microgrid virtual inertia model, Section 3 presents the RL control method and case study, and Section 4 includes results, algorithm comparison, and sensitivity analysis. Finally, the conclusions are presented.

2. Fundamental Concept and Structure of Reinforcement Learning

The true action-value associated with a is denoted by $q(a)$, whereas its empirical estimate at iteration t th time step as $Q_t(a)$. If by the t th time step action a has been chosen $N_t(a)$ times prior to t , yielding rewards $R_1, R_2, \dots, R_{N_t(a)}$, then its value is estimated according to the following Equation (1) [34]:

$$Q_t(a) = \frac{R_1 + R_2 + \dots + R_{N_t(a)}}{N_t(a)} \quad (1)$$

The simplest action selection rule is to select the action (or one of the actions) with highest estimated action value, that is, to select at step t one of the greedy actions, A_t^* , for which $Q_t(A_t^*) = \max_a Q_t(a)$. This greedy action selection method can be written as Equation (2):

$$A_t = \arg \max_a Q_t(a) \quad (2)$$

where $\arg \max_a$ denotes the value of a at which the expression that follows is maximized (with ties broken arbitrarily).

For some action, let Q_k denote the estimate for its k th reward, that is, the average of its first $k - 1$ rewards. Given this average and a k th reward for the action, R_k , then the average of all k rewards can be computed by

$$\begin{aligned} Q_{k+1} &= \frac{1}{k} \sum_{i=1}^k R_i \\ &= \frac{1}{k} \left(R_k + \sum_{i=1}^{k-1} R_i \right) \\ &= \frac{1}{k} (R_k + (k-1)Q_k + Q_k - Q_k) \\ &= \frac{1}{k} (R_k + kQ_k - Q_k) \\ &= Q_k + \frac{1}{k} [R_k - Q_k]. \end{aligned} \quad (3)$$

which holds even for $k = 1$, obtaining $Q_2 = R_1$ for arbitrary Q_1 . This implementation requires memory only for Q_k and k , and only the small computation of (3) for each new reward [34]. In nonstationary environments, recent rewards are weighted more using a constant step-size instead of uniform averaging. Thus, the incremental update (3) for Q_k based on $k - 1$ rewards is modified using a constant step-size parameter.

$$Q_{k+1} = Q_k + \alpha [R_k - Q_k], \quad (4)$$

$$\begin{aligned}
Q_{k+1} &= Q_k + \alpha[R_k - Q_k] \\
&= \alpha R_k + (1 - \alpha)Q_k \\
&= \alpha R_k + (1 - \alpha)[\alpha R_{k-1} + (1 - \alpha)Q_{k-1}] \\
&= \alpha R_k + (1 - \alpha)\alpha R_{k-1} + (1 - \alpha)^2 Q_{k-1} \\
&= \alpha R_k + (1 - \alpha)\alpha R_{k-1} + (1 - \alpha)^2 \alpha R_{k-2} + \dots \\
&\quad + (1 - \alpha)^{k-1} \alpha R_1 + (1 - \alpha)^k Q_1 \\
&= (1 - \alpha)^k Q_1 + \sum_{i=1}^k \alpha (1 - \alpha)^{k-i} R_i.
\end{aligned} \tag{5}$$

where the step-size parameter $\alpha \in (0, 1]$ is constant. This results in Q_{k+1} being a weighted average of past rewards and the initial estimate Q_1 .

Modeling of Multiple Cluster Virtual Inertia Control

As an initial step of the study, a low-order model in the Laplace domain was established to emulate the generation–load dynamics. The system comprises two area controllers interconnected through transmission lines. Solar generation in Microgrid 1 and wind generation in Microgrid 2 are considered as electrical disturbances. This configuration leads to reduced inertia and, consequently, low frequency stability. The primary physical limitation of thermal generation lies in the allowable rate of power variation, which is constrained by mechanical motion limits. These dynamics can be represented by the turbine valve-gate limits, namely the maximum and minimum opening/closing rates, together with the generation rate constraint (GRC). The parameters V_U and V_L denote the upper and lower bounds that govern the valve-gate actuation speed of the turbine unit [30]. In this study, the governor dead band of the non-reheat thermal generating units is considered as 20% p.u. MW/min for Area 1 and 25% p.u. MW/min for Area 2.

In addition to the interconnected system, the area control error (ACE) is formulated as a linear combination of the weighted frequency deviation and the tie-line power flow, as expressed below [28]:

$$\Delta P_{Cj} = \frac{K_{SI}}{s} [\beta_j \cdot \Delta f_j] + \Delta \tag{6}$$

The dynamic formulation of tie-line power is detailed in [35]. Alternatively, for an interconnected power system with N areas, the variation in tie-line power between area j and the remaining areas can be expressed as follows [25,36]:

$$\Delta P_{Tie,j}(s) = \sum_{\substack{j=1 \\ j \neq k}}^y \Delta P_{Tie,jk}(s) \tag{7}$$

$$= \frac{2\pi}{s} \left[\sum_{\substack{j=1 \\ j \neq k}}^y T_{jk} \Delta f_j - \sum_{\substack{j=1 \\ j \neq k}}^y T_{jk} \Delta f_k \right] \tag{8}$$

$$\Delta P_{Tie,jk} = \alpha_{kj} \Delta P_{Tie,kj} \tag{9}$$

$$\Delta \alpha_{kj} = -\frac{P_{rk}}{P_{rj}} \tag{10}$$

where T_{jk} denotes the synchronizing coefficient between interconnected areas, while $\Delta P_{Tie,jk}$ represents the tie-line power exchange between area j and area k . The parameters P_{rk} and P_{rj} correspond to the rated power of areas k and j , respectively.

By linearizing the load–generation dynamics—while accounting for virtual inertia as well as primary and secondary control loops, as illustrated in Figure 2—the frequency deviation in Control Area 1 can be formulated as in [30].

$$\Delta f_1(s) = \frac{1}{2H_1s + D_1}(\Delta P_{m1}(s) + \Delta P_{PV1}(s) + \Delta P_{VI1}(s) - \Delta P_{L1}(s) - \Delta P_{Tie,1}(s)) \quad (11)$$

where ΔP_m denotes the change in power generated by the thermal unit, while ΔP_g represents the turbine output signal. The term ΔP_{VI} corresponds to the emulated power from the virtual inertia control. Moreover, ΔP_W and ΔP_{PV} indicate the power variations from the wind and solar units, respectively, and ΔP_C represents the secondary control signal [30]. Equations (12) to (17) are detailed below for each of the parameters.

$$\Delta P_{m1}(s) = \frac{1}{1 + sT_{t1}} \Delta P_{g1}(s) \quad (12)$$

$$\Delta P_{g1}(s) = \frac{1}{1 + sT_{g1}} \left(\Delta P_{c1}(s) - \frac{1}{R_1} \Delta f_1(s) \right) \quad (13)$$

$$\Delta P_{c1}(s) = \frac{K_{s1}}{s} (\beta_1 \Delta f_1(s)) + \Delta P_{Tie,1} \quad (14)$$

$$\Delta P_{PV1}(s) = \frac{1}{1 + sT_{PV1}} \Delta P_{solar}(s) \quad (15)$$

$$\Delta P_{VI1}(s) = \frac{sK_{VI1} + D_{VI1}}{1 + sT_{INV1}} \left(\frac{\Delta f_1(s)}{R_{VI1}} \right) \quad (16)$$

$$\Delta P_{Tie,1} = \Delta P_{Tie,12} = T_{12} \cdot \frac{2\pi}{s} (\Delta f_1 - \Delta f_2) \quad (17)$$

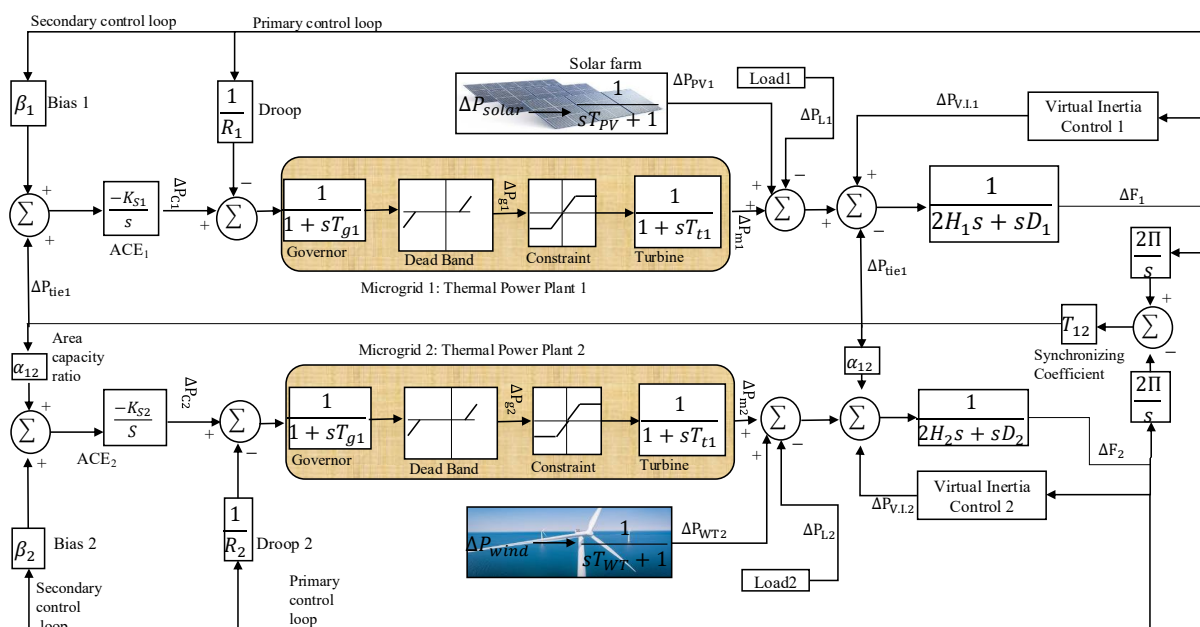


Figure 2. Microcluster integration with virtual inertia.

3. Multiple Virtual Inertia Control and Methods

The proposed workflow Figure 3 describes a data-driven control framework based on the DDPG algorithm to enhance frequency stability in microgrid clusters by minimizing ROCOF and frequency nadir.

Initially, the microgrid cluster, composed of distributed energy resources (DERs), is subjected to power disturbances such as renewable intermittency and load variations. These disturbances produce frequency deviations, which are quantified in the frequency deviation analysis block and serve as the main state inputs to the learning process. Subsequently, The core of the framework is the reinforcement learning layer, where a DDPG agent operates in a continuous action space. The agent follows an actor–critic architecture: the actor generates control actions, namely the optimal virtual inertia H_v and damping coefficient D_v , while the critic evaluates these actions through a value function. Thus, these control signals are embedded into the system dynamics by replacing the conventional transfer function with the swing-equation-based model:

$$H_v \dot{\omega}(t) + D_v [\omega(t) + \Delta\omega(t)] = -\Delta P(t) \quad (18)$$

which enables real-time adaptation of inertia and damping in response to system disturbances.

The environment simulation block represents the microgrid dynamic model, where the agent interacts iteratively. At each time step, the system returns updated frequency states and performance metrics, including ROCOF and frequency nadir.

The reward function plays a critical role in guiding the learning process. It is formulated to penalize large frequency deviations and high ROCOF values, while also regularizing excessive control effort:

$$r = -\left(w_f |\Delta f| + w_r |\text{ROCOF}| + w_u (H_v^2 + D_v^2)\right), \quad (19)$$

where w_f , w_r , and w_u are weighting factors. These adjusted values are presented in the following Table 1

Table 1. Function Reward parameter values.

Description	Symbol	Parameter Value
Frequency error weight	w_f	10
ROCOF weight	w_r	8
Control magnitude penalty	w_u	0.05

The reward weights were selected to prioritize frequency stability while maintaining smooth and physically feasible control actions. The frequency error weight w_f gives higher importance to minimizing frequency deviations, whereas the ROCOF weight w_r penalizes abrupt frequency variations and improves transient behavior. In contrast, w_u limits excessive virtual inertia and damping values, preventing unrealistic control magnitudes. Therefore, the selected weighting structure balances frequency regulation, ROCOF reduction, tie-line oscillation damping, and control effort regularization.

During the agent training phase, the DDPG agent iteratively improves its policy, learning coordinated control strategies across the microgrid cluster. This results in coordinated frequency response, improved system stability, and adaptive inertia control under varying operating conditions.

Finally, the optimized control policy leads to enhanced frequency stability, characterized by reduced ROCOF peaks and improved frequency nadir, ensuring robust operation of the microgrid cluster under high renewable penetration.

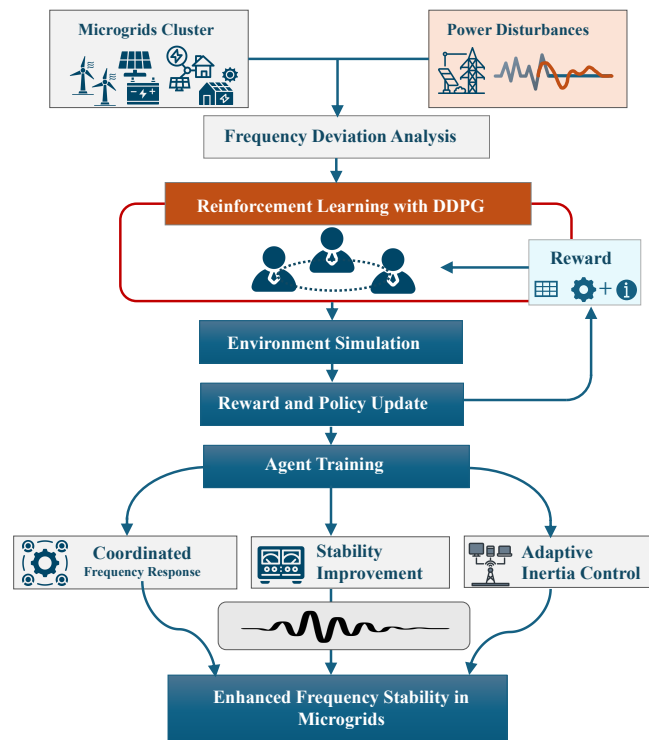


Figure 3. Proposed workflow for frequency stability enhancement in microgrid clusters using Reinforcement Learning.

Once the workflow has been defined, the implementation proceeds with the construction of the corresponding blocks in the Simulink environment. The first step consists of designing the reinforcement learning agent, which represents the core decision-making component of the control framework. According to the literature, a variety of agent architectures can be employed, depending on the nature of the action space and the control objectives. These include the Table 2.

Table 2. Table lists agents in increasing order of complexity.

Discrete Actions	Value-Based	Continuous Action Space	Actor-Critics
Q-Learning	rlQAgent	Deep Deterministic Policy Gradient (DDPG)	rlDDPGAgent
SARSA	rlSARSAAgent	Twin-Delayed Deep Deterministic Policy Gradient (TD3)	rlTD3Agent
Deep Q-Network (DQN)	rlDQNAgent	Proximal Policy Optimization (PPO)	rlPPOAgent
Proximal Policy Optimization (PPO)	rlPPOAgent	Soft Actor-Critic (SAC)	rlSACAgent
Trust region policy optimization (TRPO)	rlTRPOAgent	Trust region policy optimization (TRPO)	rlTRPOAgent

Case Study

In this case study, a dynamic analysis of a solar–wind–diesel system is conducted, incorporating energy storage systems operating in islanded mode and accounting for disturbances introduced by renewable energy sources. The mathematical modeling, along with the parameters for virtual inertia control, is derived from the characteristics of the generation units and systems available at the Centro Científico, Tecnológico y de Investigación Balzay (CCTI-B) of the University of Cuenca [37,38]. The system configuration is illustrated in [7]. The laboratory setup includes a photovoltaic system (35 kW), a mini wind farm (15 kW), a diesel generator (40 kVA), and energy storage based on a Valve-Regulated Lead-Acid (VRLA) gel battery bank (15 kWh). Additionally, a programmable load enables the creation of step disturbances in increments of 5 kW. All components are interconnected

at a point of common coupling (PCC) on the bus, operating under islanded conditions, as described below.

The next step involves establishing the workflow. As an initial stage, a block diagram was developed, which remains scarcely addressed in the literature and constitutes a primary contribution of the authors. Based on Figure 4, and taken as parameter for simulink of Table 3 the reinforcement learning control is incorporated by introducing the reward block. Additionally, the transfer function of the block representing the inertia constant H and damping factor D is replaced with its corresponding differential equation, enabling the integration of the updated virtual inertia H_v and damping coefficient D_v generated by the learning agent.

Table 3. System and Control parameter values.

Description	Microgrid 1	Microgrid 2
Thermal Power Station T_g	0.08	0.06
Thermal Power Station (Turbine) T_t	0.4	0.44
Maximum limit of valve gate	0.4	0.5
Area capacity ratio between two areas, α_{12}	−0.6	−
Wind Generation Unit T_{WT}	−	1.5
Solar Generation Unit T_{PV}	1.3	−
Primary control loop—Droop 1/ R	3	2.73
Secondary control loop—Bias	0.3483	0.3827
Synchronizing coefficient, T_{12}	0.08	−
Area control error K_S	0.3	0.2
Inertia constant H	0.083	0.1010
Inverter-based ESS	± 0.2	± 0.3

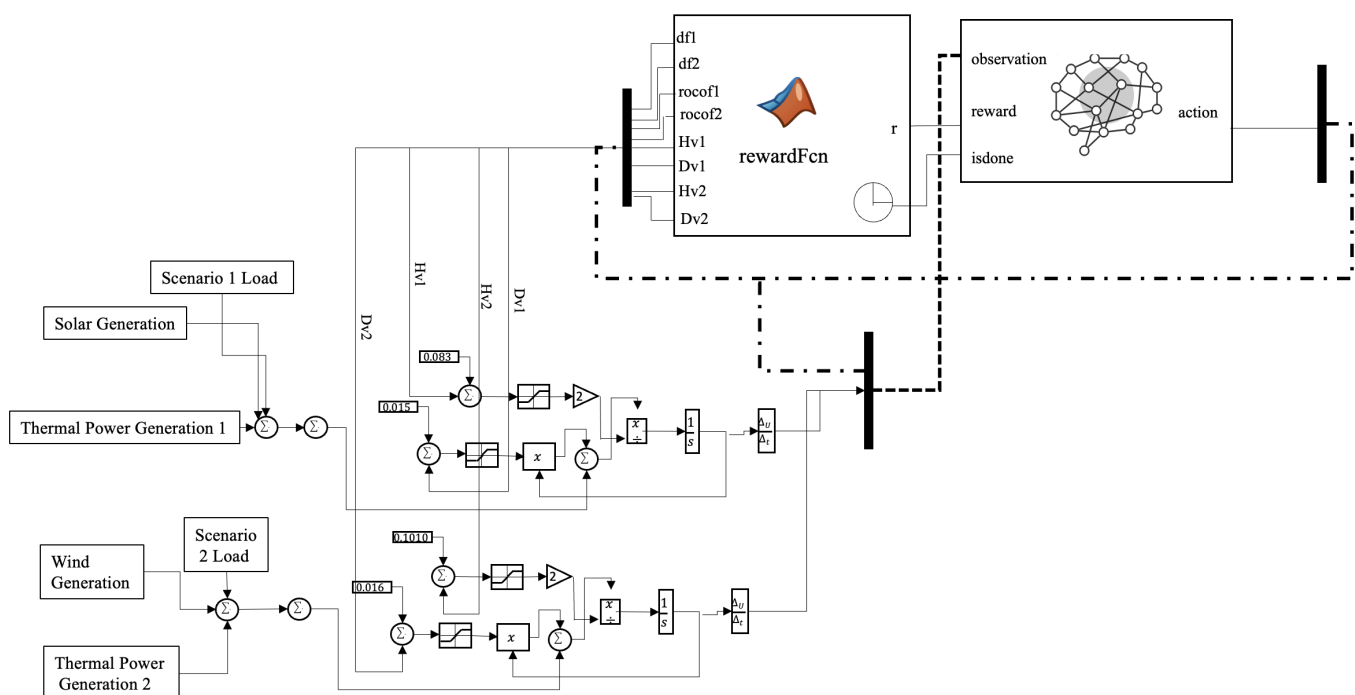


Figure 4. Schematic representation of the inertia estimation model using a reinforcement learning DDPG agent.

The Algorithm 1 will be defined as the foundation of the framework for constructing the `r1DDPGAgent`. Prior to this stage, it is necessary to formulate the observation-related algorithms, since the observation space provides the input variables required by the agent for interaction with the environment and subsequent policy learning.

Algorithm 1 Algorithm for Observation and Action Specifications

- 1: Set the number of observations: $numObs \leftarrow 5$
- 2: Set the number of actions: $numAct \leftarrow 4$
- 3: Define the observation space:
 $obsInfo \leftarrow rlNumericSpec([numObs \ 1],$
 $\quad LowerLimit = [-1; -1; -2; -2; -1],$
 $\quad UpperLimit = [1; 1; 2; 2; 1])$
- 4: Assign the observation name:
 $obsInfo.Name \leftarrow \text{"observations"}$
- 5: Define the action space:
 $actInfo \leftarrow rlNumericSpec([numAct \ 1],$
 $\quad LowerLimit = [0; 0; 0; 0],$
 $\quad UpperLimit = [1; 1; 1; 1])$
- 6: Assign the action name:
 $actInfo.Name \leftarrow \text{"actions"}$
- 7: Define the RL agent block:
 $agentBlk \leftarrow mdl + \text{" /RL Agent"}$
- 8: Create the Simulink reinforcement learning environment:
 $env \leftarrow rlSimulinkEnv(mdl, agentBlk, obsInfo, actInfo)$
- 9: Define the sampling time:
 $T_s \leftarrow 0.05$
- 10: Define the final simulation time:
 $T_f \leftarrow 20$

Next, the Algorithm 2 for constructing the actor is defined. This component is responsible for generating the control action based on the observed system states, thereby enabling the agent to interact with the environment and learn an appropriate policy for frequency stability enhancement.

Algorithm 2 Actor Network Definition

- 1: Define the actor neural network:
 $actorNet \leftarrow [$
 $\quad featureInputLayer(numObs, Normalization = \text{"none"}, Name = \text{"state"})$
 $\quad fullyConnectedLayer(128, Name = \text{"fc1"})$
 $\quad reluLayer(Name = \text{"relu1"})$
 $\quad fullyConnectedLayer(128, Name = \text{"fc2"})$
 $\quad reluLayer(Name = \text{"relu2"})$
 $\quad fullyConnectedLayer(64, Name = \text{"fc3"})$
 $\quad reluLayer(Name = \text{"relu3"})$
 $\quad fullyConnectedLayer(numAct, Name = \text{"fc4"})$
 $\quad sigmoidLayer(Name = \text{"sigmoidOut"})$
 $\quad]$
- 2: Create the deterministic continuous actor:
 $actor \leftarrow rlContinuousDeterministicActor(actorNet, obsInfo, actInfo)$

Subsequently, once the observation and actor Algorithm 3 have been established, the critic algorithm is defined. This stage is essential, as the critic evaluates the quality of the actions selected by the actor by estimating the expected return, thereby guiding the learning process toward an improved control policy.

Algorithm 3 Critic Network Definition

```

1: Define the state input path:
   statePath ← [
       featureInputLayer(numObs, Normalization = "none", Name = "state")
       fullyConnectedLayer(128, Name = "stateFC1")
       reluLayer(Name = "stateRelu1")
       fullyConnectedLayer(128, Name = "stateFC2")
   ]
2: Define the action input path:
   actionPath ← [
       featureInputLayer(numAct, Normalization = "none", Name = "action")
       fullyConnectedLayer(128, Name = "actionFC1")
   ]
3: Define the common critic path:
   commonPath ← [
       additionLayer(2, Name = "add")
       reluLayer(Name = "commonRelu1")
       fullyConnectedLayer(64, Name = "commonFC2")
       reluLayer(Name = "commonRelu2")
       fullyConnectedLayer(1, Name = "qOut")
   ]
4: Initialize the critic layer graph:
   criticNet ← layerGraph()
5: Add the state path to the critic network:
   criticNet ← addLayers(criticNet, statePath)
6: Add the action path to the critic network:
   criticNet ← addLayers(criticNet, actionPath)
7: Add the common path to the critic network:
   criticNet ← addLayers(criticNet, commonPath)
8: Connect the state path to the addition layer:
   criticNet ← connectLayers(criticNet, "stateFC2", "add/in1")
9: Connect the action path to the addition layer:
   criticNet ← connectLayers(criticNet, "actionFC1", "add/in2")
10: Create the critic Q-value function:
    critic ← rlQValueFunction(criticNet, obsInfo, actInfo)

```

The next step is to define the Algorithm 4 for agent generation by integrating the previously established observation, actor, and critic structures into a unified reinforcement learning framework. Finally, the agent (Algorithm 5) is trained over 600 episodes, which represents a conservative choice that balances computational cost and convergence performance, while still providing sufficient interaction for the learning process to stabilize.

Algorithm 4 Agent

```

1: agentOpts = rlDDPGAgentOptions;
2: agentOpts.SampleTime = Ts;
3: agentOpts.DiscountFactor = 0.99;
4: agentOpts.TargetSmoothFactor =  $1 \times 10^{-3}$ ;
5: agentOpts.MiniBatchSize = 256;
6: agentOpts.ExperienceBufferLength =  $1 \times 10^6$ ;
7: agentOpts.NoiseOptions.Variance = 0.05;
8: agentOpts.NoiseOptions.VarianceDecayRate =  $1 \times 10^{-5}$ ;
9: agent = rlDDPGAgent(actor, critic, agentOpts);

```

Algorithm 5 Training Procedure

- 1: Define the training options:
 $\text{trainOpts} \leftarrow \text{rlTrainingOptions}(\text{MaxEpisodes} = 600, \text{MaxStepsPerEpisode} = \lceil T_f/T_s \rceil, \text{ScoreAveragingWindowLength} = 20, \text{Verbose} = \text{true}, \text{Plots} = \text{"training-progress"})$
- 2: Train the reinforcement learning agent:
 $\text{trainingStats} \leftarrow \text{train}(\text{agent}, \text{env}, \text{trainOpts})$

4. Results and Discussion

The proposed deep reinforcement learning framework was evaluated in a microgrid cluster environment under diverse frequency disturbance scenarios. The main findings are summarized as follows:

The application of the reinforcement learning algorithm to virtual inertia control is depicted in Figure 5, the frequency response becomes smoother compared with classical droop control methods. This smoother behavior reduces abrupt variations in the inverter power-control signal, improving its dynamic performance under disturbance conditions. The ROCOF recorded in Area 1 is 0.1484 Hz/s, while in Area 2 it is 0.0125 Hz/s. Additionally, the frequency nadir occurs at 100.92 s in Area 1 and at 95.72 s in Area 2, indicating the dynamic response of each microgrid area under the proposed RL-based virtual inertia control strategy.

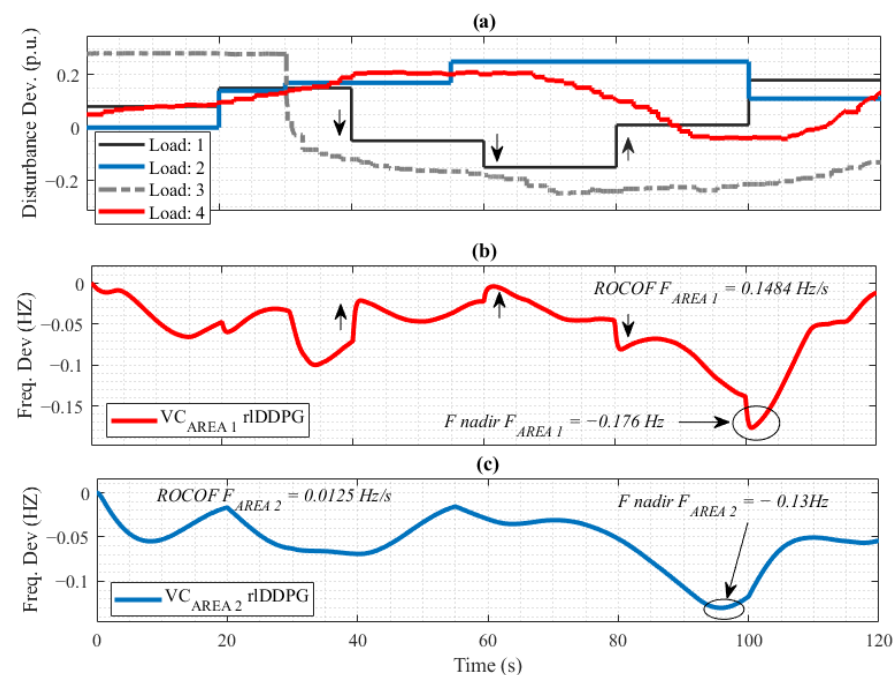


Figure 5. Improved ROCOF and frequency nadir response using a DDPG-based reinforcement learning algorithm: (a) Mixed-load disturbances, (b) Frequency derivation in Area 1 and (c) Frequency derivation in Area 2.

Figure 6 compares the dynamic performance of the PID controller and the reinforcement learning-based virtual inertia control under mixed-load disturbances. In Area 1, the PID controller reaches a ROCOF of 1.2065 Hz/s and a frequency nadir of -0.3083 Hz. In contrast, the RL-based controller provides a smoother frequency trajectory, reducing abrupt oscillations and improving the transient response.

For Area 2, the PID controller records a ROCOF of 0.6944 Hz/s and a frequency nadir of -0.1883 Hz. The RL-based control also shows a more damped response in this area, with lower oscillatory behavior and a less severe frequency deviation. These results indicate that the reinforcement learning strategy improves frequency stability by reducing both the ROCOF and the depth of the frequency nadir compared with the classical PID controller.

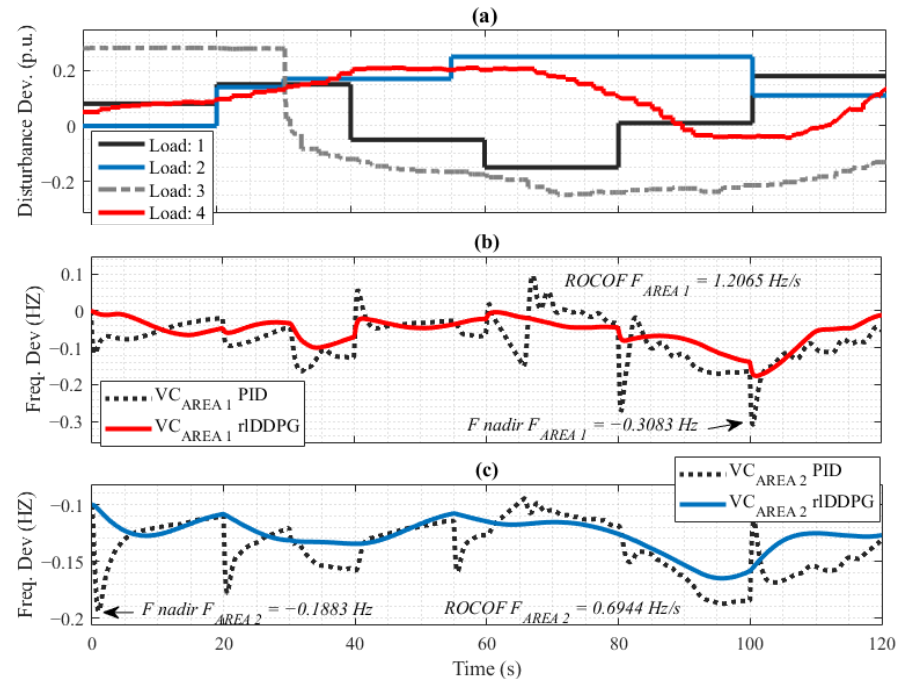


Figure 6. Comparative performance of PID and reinforcement learning-based virtual inertia control in terms of ROCOF and frequency nadir under mixed-load disturbances: (a) Mixed-load disturbances, (b) Frequency derivation in Area 1 and (c) Frequency derivation in Area 2.

As a result, once each of these algorithms has been implemented within a MATLAB R2024b script, the agent training process is carried out. The outcome of this procedure can be illustrated in the following Figure 7, which presents the training performance obtained for the proposed reinforcement learning agent. Consequently, these results confirm that the proposed model enhances both the resilience and efficiency of microgrid clusters, providing a promising pathway for future deployment in renewable-dominated power systems.

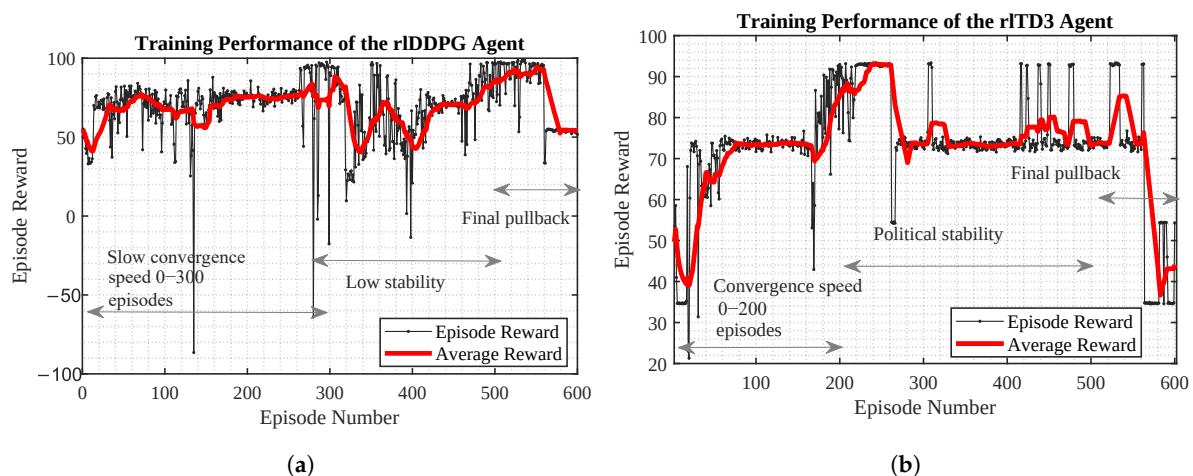


Figure 7. Cont.

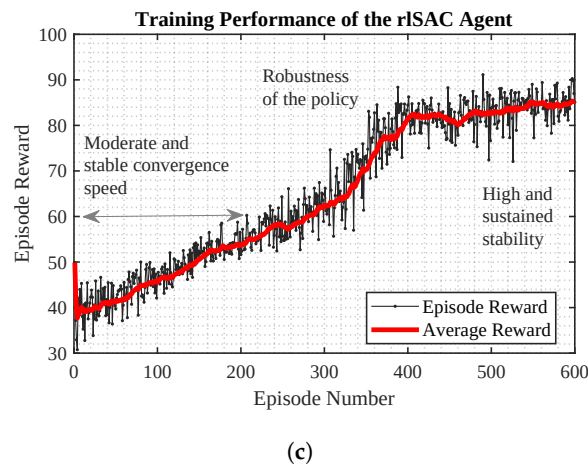


Figure 7. Training performance comparison of the reinforcement learning algorithms: (a) rLDDPG Agent, (b) rLTD3 Agent and (c) rLSAC Agent.

4.1. Performance Comparison Using Different Reinforcement Learning Algorithms

A sensitivity analysis is performed using the reinforcement learning algorithms listed in Table 4. This analysis allows evaluating the dynamic performance of different agents under the same virtual inertia control framework. In addition, the evaluation begins with the Twin-Delayed Deep Deterministic Policy Gradient (TD3) algorithm.

Table 4. Experimental Results: DDPG vs TD3 vs SAC over 600 Episodes.

Description	Deep Deterministic Policy Gradient (DDPG)	Twin-Delayed Deep Deterministic (rLTD3Agent)	Soft Actor-Critic (SAC)
Episode number	600	600	600
Episode reward	51.7044	54.3124	85.0978
Episode steps	400	400	400
Total agent steps	240,000	240,000	240,000
Average rewards	53.87	44.1195	85.3887
Average steps	400	400	400
Episode Qo	17.8135	5.4927	11.0785
Averaging windows length	20	20	20
Training stopped by	MaxEpisodes	MaxEpisodes	MaxEpisodes
Training stopped at	Episode 600	Episode 600	Episode 600
Convergence	Episode 600	No	Yes

The DDPG agent shows learning capability Figure 7a, the training process, reaching high episode reward values in several training intervals. However, the reward curve presents significant oscillations and abrupt drops, which indicate unstable policy behavior. Although the agent achieves rewards close to 90–100 between episodes 480 and 540, the final average reward decreases to 53.87. Therefore, the final trained policy does not represent the best performance obtained during training, and the best saved agent should be selected based on the maximum average reward rather than the last episode.

Although TD3 Figure 7b, reaches high reward values during training, the final training stage shows instability. Therefore, additional tuning of the exploration noise, learning rates, and reward function is required before considering TD3 as the best controller for virtual inertia optimization. Besides, as shown Figure 7c, the SAC agent presents a progressive and stable learning behavior during the 600 training episodes. The episode reward increases from approximately 40 at the beginning of training to values close to 85 at the final stage. In addition, the average reward reaches 85.3887, indicating that the learned policy maintains consistent performance under the considered operating conditions. Unlike the TD3 agent, SAC does not exhibit a significant degradation at the end of training, which suggests better policy robustness for the proposed virtual inertia control application.

4.2. Sensitivity Analysis of the Four Methods with Respect to ROCOF and Frequency Nadir

Sensitivity analysis provides a systematic evaluation of how different control methods respond to variations in system disturbances. As shown in Figures 8 and 9 for areas 1 and 2 of the cluster microgrid respectively, the classical PID-based virtual inertia control method exhibits lower performance compared with the proposed reinforcement learning-based approaches implemented using three different algorithms. In particular, the DDPG algorithm slightly outperforms, by approximately one thousandth, the agents trained with SAC- and TD3-based actor architectures in terms of frequency nadir. During the simulation, the latter achieved a frequency nadir deviation of 0.1763 Hz (See Fig 8b).

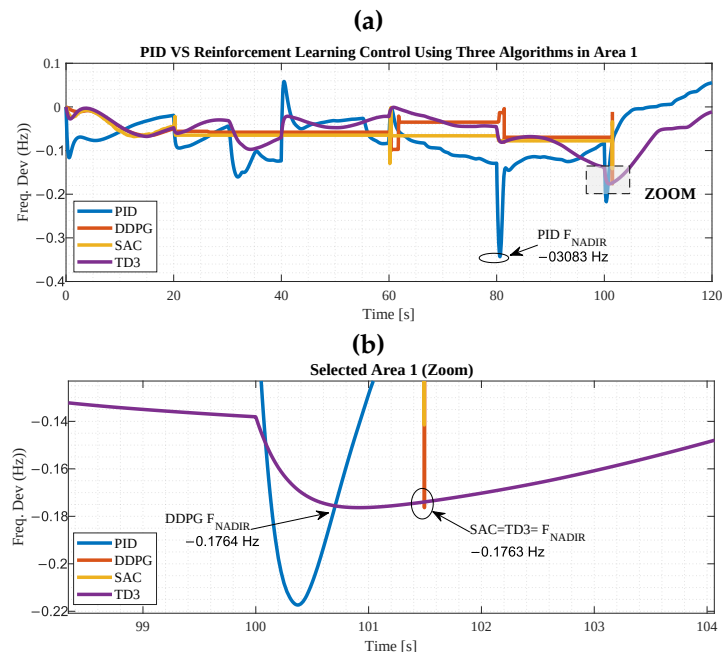


Figure 8. Sensitivity analysis in Area 1 microgrid cluster: (a) Overall frequency response and (b) Detailed view of the selected nadir region.

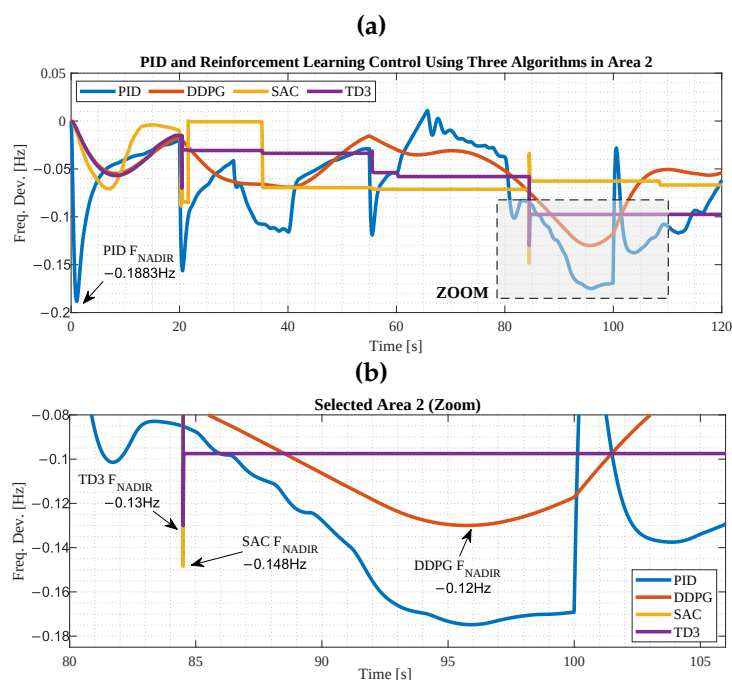


Figure 9. Sensitivity analysis in Area 2 microgrid cluster: (a) Overall frequency response and (b) Detailed view of the selected nadir region.

4.3. Training Convergence Comparison Using Moving Average Reward

The convergence behavior is presented in Figure 10, where the moving-average reward of the three reinforcement learning agents, DDPG, TD3, and SAC, is compared. The SAC agent exhibits the most stable learning trend, with a progressive increase from approximately 40 reward units at the beginning of training to nearly 85–86 reward units in the final episodes. This behavior suggests a smooth and consistent convergence process, particularly after approximately 400 episodes.

In contrast, the DDPG agent achieves high reward values, close to 90–95 reward units, between episodes 500 and 560. However, its performance decreases toward the end of training, reaching approximately 55 reward units, which indicates reduced stability in the final stage. Similarly, the TD3 agent shows a rapid improvement during the first 250 episodes, reaching nearly 93 reward units, but its reward later decreases and finishes around 43–45 reward units. Therefore, although DDPG and TD3 achieve higher transient reward peaks, the SAC agent provides the most consistent convergence behavior and the best overall training stability.

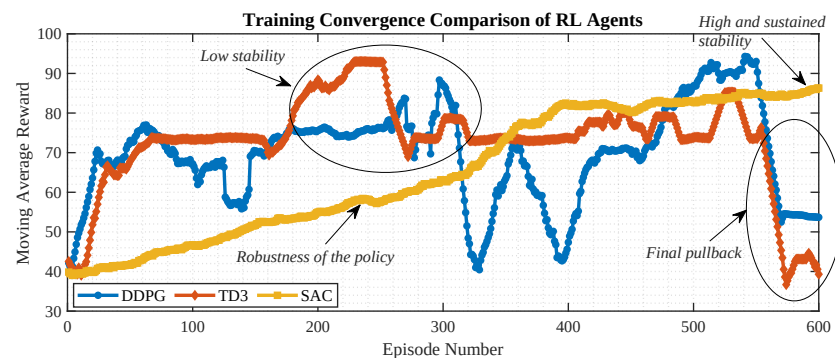


Figure 10. Comparison of Convergence Trends among DDPG, TD3, and SAC.

Table 5 summarizes the main qualitative characteristics of three reinforcement learning deep learning algorithms: TD3, DDPG, and SAC. It highlights aspects such as convergence speed, training stability, average reward, policy robustness, and exploratory behavior.

Table 5. Comparison among reinforcement learning agents.

Aspect	Twin-Delayed Deep Deterministic (TD3)	Deep Deterministic Policy Gradient (DDPG)	Soft Actor-Critic (SAC)
Convergence speed	Fast (before 200 episodes)	Slow, starting from negative values	Moderate, but stable
Stability	High between 200–500, then decline	Low, with fluctuations and drops	High and sustained
Average reward achieved	~80–90, with final setback	~40–60, with variability	~85–95, stable
Policy robustness	Good, but sensitive at the end	Lower, affected by noise and exploration	Very good, consistent
Exploratory behavior	Controlled, though with setback	Persistent, even in late phases	Balanced, with efficient exploration

5. Conclusions

This article presents a model for improving virtual inertia control in microgrid clusters using a frequency response model based on reinforcement learning. The proposed framework demonstrates that integrating deep reinforcement learning into virtual inertia control enables robust and adaptive frequency stabilization in interconnected microgrids. By analyzing sensitivity with respect to the ROCOF and frequency nadir, the study highlights

the convergence and stability achieved using SAC (Stepping-Analysis and Coordination) compared to conventional approaches. These findings confirm that intelligent, data-driven coordination mechanisms can enhance resilience under high renewable energy penetration, offering a scalable pathway toward reliable and sustainable energy systems.

The DDPG agent reaches high reward values, close to 90–100, especially between episodes 480 and 540. However, the reward presents several abrupt drops during training, indicating policy instability. Near the final episodes, the average reward decreases significantly from values close to 90 to approximately 53.87. The final episode reward is 51.7044, which is much lower than the best reward obtained during training. The increasing Episode Q0 indicates that the critic continues learning, but the actor policy does not maintain the best control performance.

The TD3 agent learned, but the final policy should not be selected only from the last episode. It is better to save the agent with the best average reward during training, around the region where the reward is close to 90.

The reward increases progressively from approximately 40 to 85 during the 600 episodes. The average reward shows a clear positive trend, indicating stable learning. Unlike TD3, the SAC agent does not present severe reward collapse near the final episodes. After approximately episode 400, the reward remains around 80–88, showing acceptable convergence. The Episode Q0 increases smoothly, suggesting consistent critic learning.

Subsequent investigations will emphasize the experimental verification of the developed methodology via hardware-in-the-loop and real-time simulation environments. Furthermore, upcoming efforts will examine the impact of communication latency, measurement inaccuracies, cybersecurity vulnerabilities, and various coordination limitations between decentralized agents. Future exploration will likewise broaden this control architecture to encompass extensive multi-area systems and assess the incorporation of sophisticated storage solutions, electric mobility, and dynamic management protocols. Such advancements will facilitate the transition of RL-driven virtual inertia strategies toward field implementation in low-inertia grids.

Author Contributions: A.C.: Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization, Visualization., D.B.: Visualization, Supervision, Software, Resources, Writing—Original Draft, Methodology, Project administration, Funding acquisition. P.A.-C.: Visualization, Supervision, Software, Resources, Writing—Original Draft, Writing—Review and Editing, Methodology. D.O.-C.: Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization, Visualization. L.I.M.-A.: Resources, Writing—Original Draft, Methodology, Investigation, Formal analysis, Data curation, Writing—Review and Editing. M.T.-V.: Resources, Writing—Original Draft, Methodology, Investigation, Formal analysis, Data curation, Writing—Review and Editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research work is part of the project titled ‘Innovative Solutions for Integrating Energy Communities into Active Distribution Networks (INNOVEC)’, with ref. PID2024-156036OB-C31, and its subproject titled ‘Optimal planning and operation of active distribution networks hosting energy communities (OptNetEC)’, funded by Ministerio de Ciencia, Innovación y Universidades, Agencia Estatal de Investigación, FEDER, UE (MCIU/AEI/10.13039/501100011033/FEDER, UE), within the Call for Knowledge Generation Projects (2024). The icons used in this document were developed by Freepik, monkik, Smashicons, and Pixel perfect, from www.flaticon.com (access 1 May 2026).

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: This manuscript is an outcome of the Doctoral Program in Renewable Natural-Resources offered by Universidad de Cuenca and Universidad del Azuay. Adrian Criollo gratefully acknowledges Universidad de Cuenca for funding his doctoral scholarship. The authors thank the Faculty of Engineering, Universidad de Cuenca, Ecuador, for easing access to the Micro-Grid

Laboratory’s facilities, allowing the use of its equipment, and authorizing its staff to provide technical support necessary to carry out the experiments described in this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RES	Renewable energy sources
PI	Proportional Integrator
PID	Proportional Integral Derivative
DDPG	Deep Deterministic Policy Gradient
SAC	Soft Actor-Critic
MADDPG	The Multi-Agent DDPG
TD3	Twin Delayed Deep Deterministic Policy Gradient
RL	Reinforcement Learning
DRL	Deep Reinforcement Learning
RLGC	Reinforcement Learning Grid Control
TDAC	Three-network Double-delay Actor-critic
PPO	Proximal Policy Optimization
MARL	Multi-agent reinforcement learning
MA-DRL	multi-agent deep reinforcement learning
CTDE	centralized training with decentralized execution
TMA-DMPG	transferred multi-agent deep meta policy gradient
EMA-DMAC	evolutionary multi-agent deep meta-actor-critic
MAAC	multi-actor-attention-critic
ROCOF	Rate of change of frequency
SMES	superconducting magnetic energy storage
EMS	Energy management system
EV	Electrical vehicle
ACE	Area control error
GRC	Generation rate constraint
VRLA	Valve-Regulated Lead-Acid
PCC	point of common coupling
PV	Photovoltaic
WT	Wind turbine
ESS	Energy storage system
BESS	Battery energy storage systems
LFC	Load frequency control
VIC	Virtual inertia controller
$q(a)$	True action-value associated with action a
$Q_t(a)$	Empirical estimate of the action-value at iteration t
$N_t(a)$	Number of times action a has been selected prior to time step t
R_k	Reward obtained at the k th selection of action a
Q_k	Estimated average reward after $k - 1$ iterations
α	Constant step-size parameter ($0 < \alpha \leq 1$)
A_t	Action selected at time step t according to the greedy policy
H_v	Virtual inertia constant determined by the actor network
D_v	Virtual damping coefficient determined by the actor network
$\omega(t)$	Instantaneous system frequency at time t
$\Delta\omega(t)$	Frequency deviation from nominal value
$\Delta P(t)$	Power imbalance at time t
r	Reward function guiding the learning process
w_f, w_r, w_u	Weighting factors for frequency deviation, ROCOF, and control effort

H_1, D_1	Inertia and damping coefficients of Area 1
ΔP_{m1}	Power variation from the thermal generating unit in Area 1
ΔP_{g1}	Turbine output signal in Area 1
ΔP_{c1}	Secondary control signal in Area 1
ΔP_{PV1}	Power variation from solar generation in Area 1
ΔP_{VI1}	Emulated power from virtual inertia control in Area 1
$\Delta P_{Tie,1}$	Tie-line power exchange between Area 1 and Area 2
T_{jk}	Synchronizing coefficient between interconnected areas j and k
P_{rj}, P_{rk}	Rated power of areas j and k respectively
K_{SI}	Integral gain of the secondary controller
β_j	Frequency bias factor for Area j
$T_{t1}, T_{g1}, T_{PV1}, T_{INV1}$	Time constants of turbine, governor, PV, and inverter dynamics
$K_{VI1}, D_{VI1}, R_{VI1}$	Virtual inertia gain, damping gain, and droop coefficient of the inverter
V_U, V_L	Upper and lower valve-gate limits of the turbine unit

References

1. Sun, X.G.; Chi, W.C.; Wang, Y.Q. Time-delay measurement and decoupling compensator control design for flexible beams under various modal excitations. *Measurement* **2025**, *257*, 118675.
2. Barbalho, P.; Moraes, A.L.; Lacerda, V.A.; Barra, P.; Fernandes, R.A.S.; Coury, D. Reinforcement Learning Solutions for Microgrid Control and Management: A Survey. *IEEE Access* **2025**, *13*, 39782–39799. <https://doi.org/10.1109/ACCESS.2025.3546578>.
3. Uddin, M.; Mo, H.; Dong, D.; Elsayah, S.; Zhu, J.; Guerrero, J.M. Microgrids: A review, outstanding issues and future trends. *Energy Strategy Rev.* **2023**, *49*, 101127. <https://doi.org/10.1016/j.esr.2023.101127>.
4. Dev, A.; Kumar, V.; Khare, G.; Giri, J.; Amir, M.; Ahmad, F.; Jain, P.; Anand, S. Advancements and Challenges in Microgrid Technology: A Comprehensive Review of Control Strategies, Emerging Technologies, and Future Directions. *Energy Sci. Eng.* **2025**, *13*, 101127. <https://doi.org/10.1002/ese3.2095>.
5. Cao, D.; Hu, W.; Zhao, J.; Zhang, G.; Zhang, B.; Liu, Z.; Chen, Z.; Blaabjerg, F. Reinforcement Learning and Its Applications in Modern Power and Energy Systems: A Review. *J. Mod. Power Syst. Clean Energy* **2020**, *8*, 101127. <https://doi.org/10.35833/mpce.2020.000552>.
6. Zhang, Z.; Zhang, D.; Qiu, R.C. Deep Reinforcement Learning for Power System: An Overview. *CSEE J. Power Energy Syst.* **2020**, *6*, 213–225. <https://doi.org/10.17775/cseejpes.2019.00920>.
7. Criollo, A.; Minchala-Avila, L.I.; Benavides, D.; Arévalo, P.; Tostado-Véliz, M.; Sánchez-Lozano, D.; Jurado, F. Enhancing virtual inertia control in microgrids: A novel frequency response model based on storage systems. *Batteries* **2024**, *10*, 18.
8. François-Lavet, V.; Henderson, P.; Islam, R.; Bellemare, M.G.; Pineau, J. An Introduction to Deep Reinforcement Learning. *Found. Trends Mach. Learn.* **2018**, *11*, 219–354. <https://doi.org/10.1561/22000000071>.
9. Yan, Z.; Xu, Y. A Multi-Agent Deep Reinforcement Learning Method for Cooperative Load Frequency Control of a Multi-Area Power System. *IEEE Trans. Power Syst.* **2020**, *35*, 4599–4608. <https://doi.org/10.1109/TPWRS.2020.2999890>.
10. Sun, X.; Qiu, J. Two-Stage Volt/Var Control in Active Distribution Networks With Multi-Agent Deep Reinforcement Learning Method. *IEEE Trans. Smart Grid* **2021**, *12*, 2903–2912. <https://doi.org/10.1109/TSG.2021.3052998>.
11. Canese, L.; Cardarilli, G.C.; Nunzio, L.D.; Fazzolari, R.; Giardino, D.; Re, M.; Spanò, S. Multi-Agent Reinforcement Learning: A Review of Challenges and Applications. *Appl. Sci.* **2021**, *11*, 4948. <https://doi.org/10.3390/app11114948>.
12. Han, Y.; Zhang, K.; Li, H.; Coelho, E.A.A.; Guerrero, J.M. MAS-Based Distributed Coordinated Control and Optimization in Microgrid and Microgrid Clusters: A Comprehensive Overview. *IEEE Trans. Power Electron.* **2018**, *33*, 6488–6508. <https://doi.org/10.1109/TPEL.2017.2761438>.
13. Dinata, N.F.P.; Ramli, M.; Jambak, M.; Megantoro, P.; Sidik, M. Simulation-Based Analysis of DRL Exploitation Rate for Voltage and Frequency Stability in a Single-Inverter Microgrid Control System. *IEEE Access* **2025**, *13*, 112731–112748. <https://doi.org/10.1109/ACCESS.2025.3580992>.
14. Khalid, J.; Ramli, M.A.M.; Khan, M.S.; Hidayat, T. Efficient Load Frequency Control of Renewable Integrated Power System: A Twin Delayed DDPG-Based Deep Reinforcement Learning Approach. *IEEE Access* **2022**, *10*, 51561–51574. <https://doi.org/10.1109/ACCESS.2022.3174625>.
15. Huang, Q.; Huang, R.; Hao, W.; Tan, J.; Fan, R.; Huang, Z. Adaptive Power System Emergency Control Using Deep Reinforcement Learning. *IEEE Trans. Smart Grid* **2020**, *11*, 1171–1182. <https://doi.org/10.1109/TSG.2019.2933191>.
16. Xi, L.; Wu, J.; Xu, Y.; Sun, H. Automatic Generation Control Based on Multiple Neural Networks With Actor-Critic Strategy. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 2483–2493. <https://doi.org/10.1109/TNNLS.2020.3006080>.

17. Altimania, M.; Alatawi, K.; Madaminov, S.; Abduvokhidov, A.; Umarov, A.; Yadav, B.K. Multiobjective Optimization Framework for Renewable Energy Integration in Smart Grids with Enhanced Stability and Resilience Using Reinforcement Learning and Distributed Control Systems. *Int. J. Low-Carbon Technol.* **2025**, *20*, 1749–1776. <https://doi.org/10.1093/ijlct/ctaf114>.
18. Li, Q.; Lin, T.; Yu, Q.; Du, H.; Li, J.; Fu, X. Review of Deep Reinforcement Learning and Its Application in Modern Renewable Power System Control. *Energies* **2023**, *16*, 4143. <https://doi.org/10.3390/en16104143>.
19. Xia, Y.; Wang, Z.; Zhao, J.; Zhang, G.; Zhang, B.; Liu, Z. A Safe Policy Learning-Based Method for Decentralized and Economic Frequency Control in Isolated Networked-Microgrid Systems. *IEEE Trans. Sustain. Energy* **2022**, *13*, 1982–1993. <https://doi.org/10.1109/TSTE.2022.3178415>.
20. Yan, R.; Wang, Y.; Xu, Y.; Dai, J. A Multiagent Quantum Deep Reinforcement Learning Method for Distributed Frequency Control of Islanded Microgrids. *IEEE Trans. Control Netw. Syst.* **2022**, *9*, 1622–1632. <https://doi.org/10.1109/TCNS.2022.3140702>.
21. Li, J.; Dai, J. Transfer Multi-Agent Deep Meta Reinforcement Learning Method for Load Frequency Control of Performance Market-Based Multi-Area Microgrid With Prosumers. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 15687–15700. <https://doi.org/10.1109/TASE.2025.3570620>.
22. Li, J.; Zhou, T. Evolutionary Multi-Agent Deep Meta Reinforcement Learning Method for Swarm Intelligence Energy Management of Isolated Multi-Area Microgrid With Internet of Things. *IEEE Internet Things J.* **2023**, *10*, 12923–12937. <https://doi.org/10.1109/JIOT.2023.3253693>.
23. Cao, D.; Zhao, J.; Hu, W.; Ding, F.; Huang, Q.; Chen, Z. Attention Enabled Multi-Agent DRL for Decentralized Volt-VAR Control of Active Distribution System Using PV Inverters and SVCs. *IEEE Trans. Sustain. Energy* **2021**, *12*, 1582–1592. <https://doi.org/10.1109/TSTE.2021.3057090>.
24. Ye, Y.; Tang, Y.; Wang, H.; Zhang, X.; Štrbac, G. A Scalable Privacy-Preserving Multi-Agent Deep Reinforcement Learning Approach for Large-Scale Peer-to-Peer Transactive Energy Trading. *IEEE Trans. Smart Grid* **2021**, *12*, 5185–5200. <https://doi.org/10.1109/TSG.2021.3103917>.
25. Norouzi, M.H.; Oshnoei, A.; Mohammadi-Ivatloo, B.; Abapour, M. Learning-Based Virtual Inertia Control of an Islanded Microgrid With High Participation of Renewable Energy Resources. *IEEE Syst. J.* **2024**, *18*, 786–795. <https://doi.org/10.1109/JSYST.2024.3370655>.
26. Frasetyo, M.B.; Wijaya, F.D.; Ali, H.R.; Wiranata, M.Z. Adaptive Synchronverter Algorithm Using Q-Learning for Microgrid Stability Control. *IEEE Access* **2025**, *13*, 53731–53747. <https://doi.org/10.1109/ACCESS.2025.3551287>.
27. Long, B.; Liao, Y.; Chong, K.T.; Rodríguez, J.; Guerrero, J.M. MPC-Controlled Virtual Synchronous Generator to Enhance Frequency and Voltage Dynamic Performance in Islanded Microgrids. *IEEE Trans. Smart Grid* **2021**, *12*, 953–964. <https://doi.org/10.1109/TSG.2020.3027051>.
28. Kerdphol, T.; Rahman, F.S.; Watanabe, M.; Mitani, Y.; Turschner, D.; Beck, H.P. Enhanced Virtual Inertia Control Based on Derivative Technique to Emulate Simultaneous Inertia and Damping Properties for Microgrid Frequency Regulation. *IEEE Access* **2019**, *7*, 14422–14433. <https://doi.org/10.1109/ACCESS.2019.2892747>.
29. Haddadi, M.; Gorji, S.A.; Yu, S.S. An Overview of Inertia Emulation Strategies for DC Microgrids: Stability Analysis and AC Microgrid Analogies. *IEEE Open J. Ind. Electron. Soc.* **2025**, *6*, 491–521. <https://doi.org/10.1109/OJIES.2025.3550625>.
30. Kerdphol, T.; Watanabe, M.; Mitani, Y.; Phunpeng, V. Applying Virtual Inertia Control Topology to SMES System for Frequency Stability Improvement of Low-Inertia Microgrids Driven by High Renewables. *Energies* **2019**, *12*, 3902. <https://doi.org/10.3390/en12203902>.
31. Konneh, K.V.; Adewuyi, O.; Lotfy, M.E.; Sun, Y.; Senjyu, T. Application Strategies of Model Predictive Control for the Design and Operations of Renewable Energy-Based Microgrid: A Survey. *Electronics* **2022**, *11*, 554. <https://doi.org/10.3390/electronics11040554>.
32. Yavuz, M.; Kıvanç, Ö.C. Optimization of a Cluster-Based Energy Management System Using Deep Reinforcement Learning Without Affecting Prosumer Comfort: V2X Technologies and Peer-to-Peer Energy Trading. *IEEE Access* **2024**, *12*, 31551–31575. <https://doi.org/10.1109/ACCESS.2024.3370922>.
33. Ding, L.; Cui, Y.; Yan, G.; Huang, Y.; Fan, Z. Distributed Energy Management of Multi-Area Integrated Energy System Based on Multi-Agent Deep Reinforcement Learning. *Int. J. Electr. Power Energy Syst.* **2024**, *157*, 109867. <https://doi.org/10.1016/j.ijepes.2024.109867>.
34. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, UK, 2021; Volume 1.
35. Kerici, T.; Hurtado, M.; Gjergji, M.; Tweed, S.; Kennedy, E.P.; Milano, F. Frequency Quality in Low-Inertia Power Systems. In *2023 IEEE Power & Energy Society General Meeting (PESGM)*; IEEE: New York, NY, USA, 2023. <https://doi.org/10.1109/PESGM52003.2023.10253411>.
36. Liu, Y.; Li, F.; Gao, L.; Zhang, Y.; Liu, Y. A Synergetic Control-Based Dynamic Frequency Control Strategy for VSG in Microgrid. In *Proceedings of the International Conference on Electrical Engineering and Control Science (IC2ECS)*; IEEE: New York, NY, USA, 2023; pp. 594–597. <https://doi.org/10.1109/IC2ECS60824.2023.10493632>.

37. Espinoza, J.; Gonzalez, L.; Sempertegui, R. Micro grid laboratory as a tool for research on non-conventional energy sources in Ecuador. In *2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*; IEEE: New York, NY, USA, 2017; pp. 1–7.
38. Ochoa, D.; Villa, E.; Iñiguez, V.; Larco, C.; Sempértégui, R. Uso de supercondensadores para brindar soporte de frecuencia en una microrred aislada. *Rev. Tecnol.-ESPOL* **2022**, *34*, 174–185.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.