energies

MDPI

Article

A Deep Reinforcement Learning Approach to DC-DC Power Electronic Converter Control with Practical Considerations

Nafiseh Mazaheri *, Daniel Santamargarita, Emilio Bueno, Daniel Pizarro and Santiago Cobreces

Department of Electronics, Alcalá University (UAH), Plaza San Diego S/N, 28801 Madrid, Spain; daniel.santamargarit@uah.es (D.S.); emilio.bueno@uah.es (E.B.); daniel.pizarro@uah.es (D.P.); santiago.cobreces@uah.es (S.C.)

* Correspondence: nafiseh.mazaheri@uah.es

Abstract: In recent years, there has been a growing interest in using model-free deep reinforcement learning (DRL)-based controllers as an alternative approach to improve the dynamic behavior, efficiency, and other aspects of DC–DC power electronic converters, which are traditionally controlled based on small signal models. These conventional controllers often fail to self-adapt to various uncertainties and disturbances. This paper presents a design methodology using proximal policy optimization (PPO), a widely recognized and efficient DRL algorithm, to make near-optimal decisions for real buck converters operating in both continuous conduction mode (CCM) and discontinuous conduction mode (DCM) while handling resistive and inductive loads. Challenges associated with delays in real-time systems are identified. Key innovations include a chattering-reduction reward function, engineering of input features, and optimization of neural network architecture, which improve voltage regulation, ensure smoother operation, and optimize the computational cost of the neural network. The experimental and simulation results demonstrate the robustness and efficiency of the controller in real scenarios. The findings are believed to make significant contributions to the application of DRL controllers in real-time scenarios, providing guidelines and a starting point for designing controllers using the same method in this or other power electronic converter topologies.

Keywords: deep reinforcement learning; proximal policy optimization; power electronic converters; buck converter

check for updates

Citation: Mazaheri, N.; Santamargarita, D.; Bueno, E.; Pizarro, D.; Cobreces, S. A Deep Reinforcement Learning Approach to DC-DC Power Electronic Converter Control with Practical Considerations. *Energies* **2024**, *17*, 3578. https://doi.org/10.3390/en17143578

Academic Editor: Ralph Kennel

Received: 17 June 2024
Revised: 13 July 2024
Accepted: 18 July 2024
Published: 21 July 2024

Copyright: © 2024 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) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Microgrids refer to a distributed network of loads and energy generation units confined within specific electrical boundaries and can be categorized into two main groups: AC and DC. DC microgrids offer several advantages over AC ones, including higher efficiency, lower cost, simplified control, protection, and smaller system size [1,2]. DC–DC power electronic converters play an important role in microgrid control and operation by facilitating efficient power transfer between different components of a microgrid by converting DC from one voltage level to another [3,4]. They are widely used in various applications such as industrial automation, aerospace, renewable energy systems, electric cars, fast EV DC chargers, and interfaces between DC catenaries and DC loads.

In these applications, it is necessary not only to optimize the temporal response but also to optimize the efficiency and nonlinear behavior of the converter. Therefore, it is essential to apply advanced control techniques such as DRL-based controllers. Moreover, these converters can operate in different modes, such as buck or boost converters that can function in DCM, CCM, or DAB converters that have multiple modulation techniques. This flexibility allows for managing different load conditions and improving performance across various applications. Among the various categories of DC–DC power electronic converters, the buck converter stands out as a popular choice due to its simplicity and high efficiency.

Numerous research studies have investigated control methods for DC–DC power electronic converters, including classical design methods such as proportional–integral–

derivative (PID) control [5], nonlinear techniques like sliding mode control (SMC) [6,7], model predictive control (MPC) [8,9], etc. Although these control methods perform well, they often rely on detailed system models. The design process of these control methods is typically based on the converter's average model, which captures only the average behavior of the converters and does not consider the switching model. For instance, in the case of dual active bridge (DAB) converters, achieving precise control and ensuring effectiveness requires consideration of the switching model, especially when integrating modulations that vary depending on the operating points [10]. Moreover, it is challenging to achieve fast and accurate control performance due to the different factors such as parameter uncertainties, nonlinearities, input voltage disturbances, and time-varying operational scenarios in complex power converter topologies.

Recent advancements in passivity-based control have shown significant improvements in the performance of power electronic systems[11]. Additionally, model-free intelligent controllers such as neural networks and fuzzy logic controllers [12–14] have been developed without requiring an explicit model of the system. Notably, the study of [15] demonstrates the effectiveness of a fuzzy logic controller in mitigating harmonics and managing reactive power in three-level shunt active filters. These controllers are suitable for specific time intervals; however, they often lack the capability for online learning.

With the rapid advancement in machine learning, DRL-based techniques, which encompass both model-based and model-free approaches, have captured significant attention and demonstrated remarkable success in addressing complex and nonlinear behaviors across various intricate problems due to their self-learning characteristics. Model-free DRL methodologies, such as the PPO algorithm, do not require a detailed mathematical model of the system they control. Instead, they learn optimal policies directly from interactions with the environment, making decisions based on trial and error to maximize cumulative rewards. This can be applied in designing complex controllers with novel functionalities for power electronic converters, particularly in scenarios where obtaining results with traditional control techniques may not be straightforward. DRL enables adaptation to converters' switching models across the entire operating range and their variations at different operating points.

Unlike traditional methods, these DRL-based methods can be used to control different configurations of DC–DC power electronic converters, including boost, buck-boost, DAB, and cascaded H-bridge (CHB) converters, simply by retraining the agents in new environments. In [16], a DRL-based optimization algorithm is proposed for optimizing the design parameters of power electronic converters using a deep neural network (DNN)-based surrogate model. It is important to note that by considering the same fundamentals of the DRL design, such as the neural network structure, reward function, training hyperparameters, and other details of the DRL methodology, the proposed method can be extended to other topologies of converters. This can be carried out by changing the environment in the Simulink model and then retraining the agent for the new environment.

In theses state-of-the-art studies, there are some works that have focused on employing DRL-based controllers for voltage control in DC–DC power electronic converters. For example, the deep deterministic policy gradient (DDPG) algorithm is utilized in [17], enabling effective control of the output voltage of the buck converter for precise tracking of a reference voltage in dynamic environments. In the study [18], the deep Q-network (DQN) algorithm is implemented for an intelligent adaptive model predictive controller, demonstrating the ability to optimize control actions considering the converter's states and environmental conditions. This method is applied to a DC–DC buck-boost converter in the presence of significant CPL variations. Ref. [19] investigates the voltage control of a buck converter using two RL agents, namely DQN and DDPG, benchmarked against controllers such as SMC and MPC. Another research suggests a PPO-based ultra-local model (ULM) scheme for achieving voltage stabilization in DC–DC buck-boost converters feeding constant power loads (CPLs) [20]. The primary focus of the work [21] is on the development of an online training artificial intelligence (AI) controller for DC–DC

converters employing the DDPG algorithm. The controller is applicable to all types of DC–DC converters and can address a variety of control objectives. In [22], an RL regression-based voltage regulator for boost converters is proposed, integrating the PWM technique. It is a model-based method requiring an accurate average model of the converter and a suitable set of kernel functions. Furthermore, in [23], an intelligent PI controller coupled with an RL-based auxiliary controller is discussed to stabilize the output voltage of a buck converter under CPL conditions. This approach is also model-based and involves manipulation of the duty cycle of the PWM signal. In [24], a novel algorithm utilizing DQN is introduced for parameter optimization, aiming to improve the design of power electronic converters. Moreover, a new online integral reinforcement learning (IRL)-based data-driven control algorithm for the interleaved DC–DC boost converter is proposed in [25]. Additionally, [26] demonstrates a variable-frequency triple-phase-shift (TPS) control strategy employing deep reinforcement learning to enhance the conversion efficiency of the dual-active-bridge converter. The ANN-RL (artificial neural networks-reinforcement learning) approach is employed in [27,28] to regulate the operation of the buck converter, showing robustness against parameter variations and load changes.

Although different DRL algorithms are proposed for power electronic converters in these state-of-the-art studies, some gaps remain in the body of knowledge regarding the impact of neural network structure on control performance and computational efficiency. Additionally, these studies do not explore the use of different input features to improve decision-making. They also lack guidelines for choosing training hyperparameters, network size, and reward function. These studies typically use an average model of the power electronic converter, which simplifies the converter's dynamics. In contrast, the proposed method in this work uses a detailed switching model that captures the complex behavior of the converter more accurately. Moreover, the performance of DC–DC power electronic converters in these studies is not considered in both modes of operations (DCM and CCM) and usually focuses on one type of load. Furthermore, the behavior of changing integral gain in the input of ANNs is also rarely studied.

Therefore, a study on the DRL training process using a switching model of DC–DC power electronic converters in different modes of operation with varying loads, along with an easily implemented DRL in real life that considers memory consumption, could be very useful. This study is not intended to perform a comparison between conventional methods and DRL controllers, as this has already been done in several studies such as [19,23,29]. These studies have shown that DRL-based controllers can control power electronic converters better than traditional controllers in complex scenarios. The main objective of this paper is to explore the potential and capabilities of model-free DRL-based controllers, particularly the PPO agent, in the control of DC–DC power electronic converters. This approach addresses existing gaps by providing a comprehensive method for enhancing control and optimization.

Our key contributions include the following:

1. Proposing a complete training guide for DRL agents to control real power electronic converters in different modes of operation using a detailed switching model that considers real-life delays. This guide also takes into account the introduction of a chattering-reduction term in the reward function and input feature selection and evaluates the impact of network size.
2. Analyzing the trade-off between neural network configuration, computational efficiency, memory usage, and control performance accuracy through experimental tests.
3. Adding an adjustable gain in the integral error input, allowing control over transient responses without the need to retrain the DRL agent.

The general control scheme of the proposed DRL controller for a buck converter is presented in Figure 1.

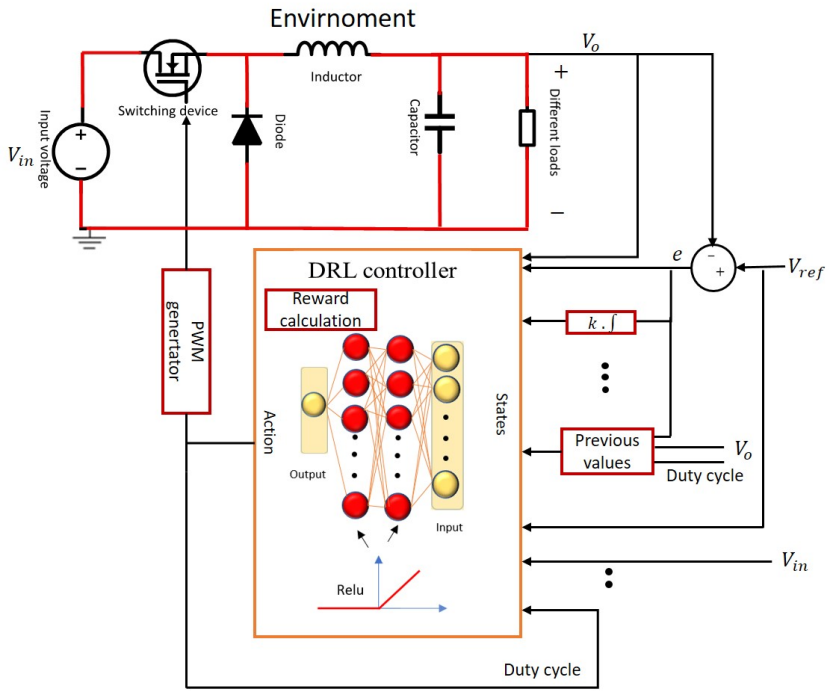


Figure 1. The proposed control scheme for a buck converter.

The rest of this paper is organized as follows. The next section provides a brief introduction to the fundamental principles of the DRL approach and the structure of the PPO algorithm. Following that, a practical case study for the buck converter is explored, implementing the DRL algorithm. The paper then presents the simulation and experimental results of the proposed algorithm in various scenarios. Finally, the conclusions drawn from the study are discussed.

2. DRL-Based Controller-Choosing Guide

DRL algorithms [30] can be broadly categorized as either model-based or model-free, as illustrated in Table 1, in which ✓✓, ✓, and - present High, Medium, and Low or None respectively. Recently, there has been a growing interest in model-free algorithms which focus on training agents to learn a near-optimal policy through iterative interactions with an environment, as shown in Figure 2. This involves receiving feedback in the form of rewards and adjusting neural network parameters to maximize cumulative discounted rewards over time, as defined in Equation (1) [30], where $\gamma \in [0, 1]$ is the discount factor applied to future rewards, π is the policy, and r_t is the reward at time step t.

$$J = \mathrm{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]. \tag{1}$$

A DRL environment is formally defined as a Markov decision process (MDP). The MDP framework outlines decision-making under the Markovian assumption, where the next state of the environment is solely influenced by the current state and the agent's action within it. An MDP is denoted as a tuple $\langle S, A, P, R, \gamma \rangle$, where S and A represent the state and action space, respectively. The transition matrix $P : S \times A \times S \to [0, 1]$ determines the probability of transitioning from one state to another when performing an action. Additionally, the matrix $R : S \times A \to \mathbb{R}$ specifies the reward of each transition [30]. In the following, each of them is described in detail.

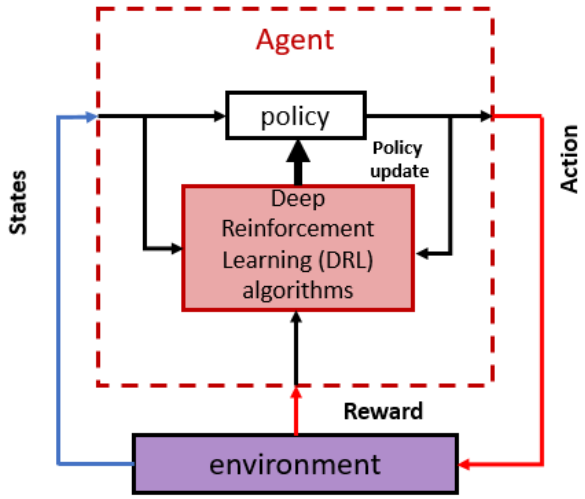


Figure 2. The general framework of DRL.

Table 1. A comparison of DRL algorithms.

Deep Reinforcement Learning Algorithms	Characteristics			
	No Need for Domain Knowledge	Sample/Computational Efficiency	Robustness to Hyper-Parameters	Support Continuous Action Space
Model-free				
Value-based (e.g., DQN)	✓	✓	✓	-
Policy-based (e.g., PG)	✓	-	-	✓
Hybrid or Actor-Critic (e.g., PPO, TD3, DDPG)	✓	✓	✓	✓
Model-based	-	✓✓	✓✓	✓

Among various model-free DRL algorithms, an actor–critic structure synergistically combines the strengths of both policy-based and value-based methods, enhancing the stability and efficiency of learning. This architecture consists of two essential components: an actor selects actions based on the current state of the environment, while a critic is typically responsible for evaluating actions and providing feedback on their effectiveness so as to achieve the desired goals. The critic is defined by another neural network. Figure 3 illustrates the rationality behind the choice of PPO as an actor–critic algorithm for controlling DC–DC power electronic converters. This choice is based on its stability, sample/computational efficiency, ability to perform online policy updates, continuous adaptation to evolving or diverse environments, and ease of implementation [31,32]. Consequently, these features make the PPO a suitable candidate for real-time control applications [33]. The PPO and its variant, PPO-Clipped [34], focus on minimizing excessive policy updates at each step, leading to a more stable training procedure.

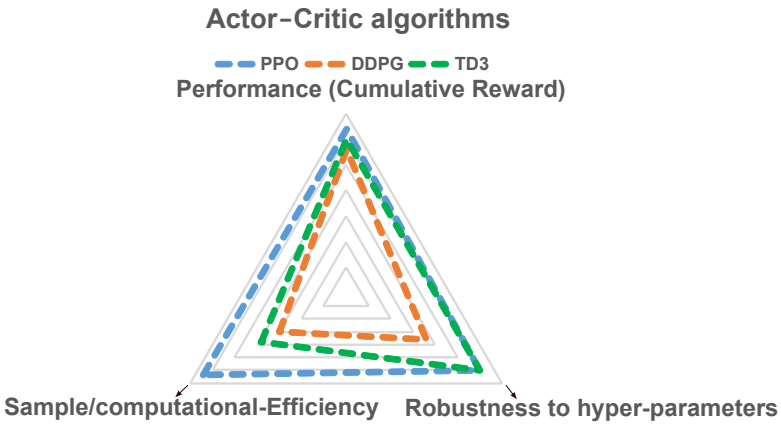


Figure 3. A comparison of selected actor–critic algorithms in terms of performance, sample/computational efficiency (i.e., number of samples/computations required to achieve convergence), and robustness to hyperparameters [30].

2.1. Fundamentals of the Design

In the design of DRL controllers for power electronic converters, it is crucial to carefully consider several fundamental aspects. These include the definition of the state space and input feature selection, action space, reward function engineering, and the architecture of the neural network. Each of these elements plays an important role in ensuring that the DRL agent can learn effectively and perform robustly when controlling the power converter.

2.1.1. State Space

The state, which contains a set of features, must contain adequate information to comprehensively describe the system environment. Hence, the selection of relevant states is vital in the design of a DRL system to improve the learning process. Apart from existing studies, additional features are considered to enable more informative decisions, representing the input information the agent receives from the environment at each time step t. Due to the consideration of various objectives and scenarios, the features encompass all time-changing variables and context parameters (refer to time-independent features), thereby enhancing the system's capabilities [35]. Remarkably, there is a trade-off between the number of previous feature values (τ) used in the states, the training time, and the size of the network. Larger values of τ lead to more efficient decisions, although this entails increased complexity. Based on extensive simulation results, this study determines that 4 is the most suitable number of previous values, reflecting a balance between effective learning and computational efficiency. Table 2 shows the feature vector S_t, where V_t, e_t, d_t, V_{in}, k, V_{ref}, and V_{max} represent the output voltage, voltage tracking error, duty cycle, input voltage, integral gain, reference voltage, and maximum output voltage, respectively. V_{max} serves as a constant for normalization. The feature vector has 19 components. Moreover, all values of the feature vector are normalized using a linear mapping technique that scales the variable between zero and one.

Table 2. The feature vector components.

Vector Components	
$\frac{V_{t-\tau}}{V_{\max}}$, for $\tau = 0,1,2,3,4$	$\frac{e_{t-\tau}}{V_{\max}}$, for $\tau = 0,1,2,3,4$
$d_{t-\tau}$, for $\tau = 0,1,2,3,4$	$\frac{V_{\text{in}}}{V_{\max}}$
$\frac{V_{\text{ref}}}{V_{\max}}$	$(d_t - d_{t-1})$
$k \int_0^t e_t \, dt$	

2.1.2. Action Space

Actions (a_t) in DRL are the decisions or control inputs. In the context of power electronic converters, the duty cycle is defined as an action by considering continuous values between 0 and 1 for the PPO method.

2.1.3. Reward Engineering

The reward function serves as a benchmark for evaluating the agent's actions within a given state in DRL. It provides the necessary feedback mechanism to inform the algorithm about the success ratio of an action, thereby establishing communication between the algorithm and the control objective. The reward function defines the goal of the control task in a quantifiable way, allowing the neural network to understand which actions are favorable.

An individual and optimized reward function is designed to reflect how well the converter is performing with regard to each crucial objective, such as output voltage regulation and minimizing actuation chattering. It is important to note that chattering refers to rapid and excessive switching between control states, leading to unpredictable behavior and unstable performance [36]. In this research work, two different components are considered in the defined reward in Equation (2), representing the different objectives and priorities. The convergence term in the reward function motivates the DRL agent to reach a stable and fast convergence rate towards the desired output, and by including the duty cycle term, the DRL agent is prompted to achieve smoother operation with less chattering in the duty cycle.

$$r_t = \begin{cases} -\beta|e_{\text{norm}}| - k \cdot e_d, & \text{if } |e_{\text{norm}}| < \varepsilon \\ -\alpha|e_{\text{norm}}| - k \cdot e_d, & \text{if } |e_{\text{norm}}| \geq \varepsilon \end{cases}, \tag{2}$$

$$e_d = (d_t - d_{t-1})^2, \quad e_{\text{norm}} = \frac{(V_t - V_{\text{ref}})}{V_{\max}}.$$

In the above equation, β and α affect the convergence speed and the final performance, ε is the maximum allowable error, and k penalizes large changes in the duty cycle, aiming to reduce actuation chattering. Determining the optimal trade-offs and right balance between the factors in each term of the rewards function in DRL is an essential task. All the mentioned notations are positive weighting coefficients and are adjusted through domain expertise or trial–error, as will be provided in Section 2.1.5 in Table 4.

2.1.4. Artificial Neural Network Design (ANN)

ANNs play a vital role in the DRL, serving as function approximators that enable the agent to learn complex patterns and make informed decisions within the DRL framework. The utilized network described in Table 3 has four layers, including an input layer, two fully connected hidden layers with 32 and 16 neurons, and an output layer. The activation function employed by each hidden layer is the rectified linear unit (ReLU) function, which is one of the most commonly used activation functions in similar studies, as indicated by [37,38]. The fully connected layers are employed to integrate the state space with the action space. The architecture of the actor and critic network (including the number of layers and neurons) can be set according to the task's complexity, the size of the input space, and the desired learning capacity [39]. Therefore, according to the trade-off analysis and the number of MACs, which will be presented in Section 4.2.4, this network size configuration is chosen to achieve a balance between performance accuracy, computational complexity, and memory usage. Moreover, the input features vector relevant to Table 2 consists of 19 components, including the output voltage, voltage tracking error, duty cycle, and the four previous values with a time step of 20 µs for each of them. Additionally, duty cycle variation, input voltage, integral gain, and reference voltage are part of the feature vector. The duty cycle is considered the action, making the ANN have one output and 19 inputs.

Table 3. PPO artificial neural network architectures.

Actor /Critic Network		
Layer	**Width**	**Activation**
Input	Number of features: 19	-
Dense (Fully-connected)	32	Relu
Dense (Fully-connected)	16	Relu
Output	1	-

2.1.5. Training Procedure

The training procedure for a PPO agent is designed to develop a policy that optimally interacts with an environment to achieve specific goals. The process begins with the agent exploring the environment, gathering state information that represents the current context and features variables, and then choosing an action based on its policy. Each action taken by the agent leads to a change in the environment, with the agent receiving a reward that indicates how favorable the outcome of that action was. The feedback from these rewards is important for learning and improving the policy. The key to PPO's success lies in its ability to balance exploration and exploitation. Exploration involves trying new actions to gather information, while exploitation uses known strategies to maximize rewards. PPO achieves this balance through a unique clipped surrogate objective function, which helps to stabilize learning by limiting drastic changes to the policy. This approach, coupled with trust region optimization, ensures that the policy updates are smooth and controlled, reducing the risk of overfitting or oscillations during training. Throughout the training process, the agent collects a set of trajectories $D_k = \{\tau_i\}$, each consisting of a sequence of states, actions, rewards, and next states. These data are used to compute advantage estimates $\hat{A}t$, which help determine how much better or worse an action is compared to a baseline value function. The agent then uses these advantage estimates to update its policy through stochastic gradient descent, adjusting its decision-making strategy to improve performance. The iterative nature of PPO's training process allows the agent to refine its policy over time, gradually increasing its ability to select actions that lead to better outcomes in the environment [40,41]. The pseudocode for the PPO-clipped algorithm, as shown in Algorithm 1, outlines the step-by-step procedure for implementing this approach [34]. In this study, PPO-clipped is implemented, andcthe terms 'PPO' and 'PPO-clipped' are used interchangeably to refer to this algorithm.

Moreover, several hyperparameters need to be tuned appropriately to achieve the best possible performance and stability during training. These parameters play a crucial role in shaping various aspects of the learning process, such as the agent's capacity to explore its environment and effectively exploit learned policies. Some of these hyperparameters are detailed in Table 4, selected based on best practices and fine-tuned to address the specific challenges of the reinforcement learning problem and environment. Notably, the learning rate determines the magnitude of updates to the neural network's weights during training. A higher learning rate may lead to faster convergence but risks overshooting the optimal policy, while a lower learning rate may result in slower convergence. To achieve a compromise between them, the value 1×10^{-4} is chosen. Similarly, the discount factor (gamma) influences the agent's consideration of future rewards. A higher discount factor encourages long-term reward prioritization, while a lower value may prioritize immediate rewards. As a result, a value of 0.9 is selected. Additionally, the entropy coefficient plays a pivotal role in balancing exploration and exploitation by incentivizing the agent to explore novel actions. A higher entropy coefficient fosters exploration, whereas a lower coefficient emphasizes the exploitation of learned policies.

Algorithm 1 PPO-clipped.

1: Input: Initial policy parameters θ_0, initial value function parameters ϕ_0
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Collect the set of trajectories $D_k = \{\tau_i\}$ by running the policy $\pi_k = \pi(\theta_k)$ in the environment.
4: Compute rewards-to-go, $\hat{r}_t$, representing the cumulative sum of future rewards from time step t onward.
5: Compute advantage estimates, $\hat{A}_t$ (using any method of advantage estimation) based on the current value function V_{ϕ_k} which represents the expected return from a given state.
6: Update the policy by maximizing the PPO-clip objective:

$$\theta_{k+1} \leftarrow \arg\max_{\theta} \frac{1}{|D_k|T} \sum_{\tau \in D_k} \sum_{t=0}^{T} \min\left(\frac{\pi_\theta(a_t|S_t)}{\pi_{\theta_k}(a_t|S_t)} \hat{A}^{\pi_{\theta_k}}(S_t, a_t), g(\epsilon, \hat{A}^{\pi_{\theta_k}}(S_t, a_t)) \right)$$

typically, via stochastic gradient ascent with Adam.
7: Fit value function by regression on mean-squared error:

$$\phi_{k+1} \leftarrow \arg\min_{\phi} \frac{1}{|D_k|T} \sum_{\tau \in D_k} \sum_{t=0}^{T} (V_\phi(s_t) - \hat{r}_t)^2,$$

typically, via some gradient descent algorithm.
8: **end for**

Table 4. DRL hyperparameters (PPO-clipped agent).

Parameter	Value
Learning rate	1×10^{-4}
Discount Factor	0.9
Entropy coefficient	0.001
Clip Factor	0.2
Rollout Size	256
$\beta, \alpha, k, \varepsilon$	1, 7, 1, 0.01
Rollout Size	256

3. A Practical Case: Implementation of the DRL Algorithm for a Buck Converter

In this study, a real buck converter is used. Despite its simplicity, which facilitates understanding, it presents challenges due to its operation between DCM and CCM, leading to significant variations in the converter model. Additionally, the consideration of two types of loads, resistive and inductive, further complicates the analysis. This serves as the environment in which the agent interacts within the framework of DRL. Figures 4 and 5 display the procedure of the PPO implementation in real-life scenarios. The architecture $(32, 16)$ is employed for the critic network, which is similar to the actor network. As illustrated in Figure 4, Python 3.9 was chosen for its full-fledged support in designing DRL algorithms, leveraging its extensive libraries and frameworks. Meanwhile, the MATLAB 2023a Simulink environment is well-known for its powerful simulation capabilities, particularly in modeling dynamic systems such as power electronic converters. Additionally, the MATLAB Engine API facilitates seamless communication between Python and MATLAB for enhanced integration and efficiency. It is valuable to note that in the MATLAB Simulink environment, the choice of model and simulation program significantly impacts the accuracy of DC–DC power electronic converter simulations [42]. In this study, the Simscape Electrical Specialized Power Systems toolbox within MATLAB/Simulink is selected due to its robustness and versatility in handling DC–DC power electronic simulations. Additionally, the use of the `PowerGui` block with a time step of 200 ns (nanoseconds) ensures that high-frequency switching events are accurately resolved, providing precise insights into the

converter's transient and steady-state behavior. Moreover, it is notable that the PPO agent operates with a time step of 20 µs (microseconds). This time step was selected because the controller was designed to operate within the constraints of the real-time simulator OPAL-RT, which also uses a time step of 20 µs (microseconds). This synchronization ensures that the control actions are applied in real-time, maintaining the intended performance of the control system.

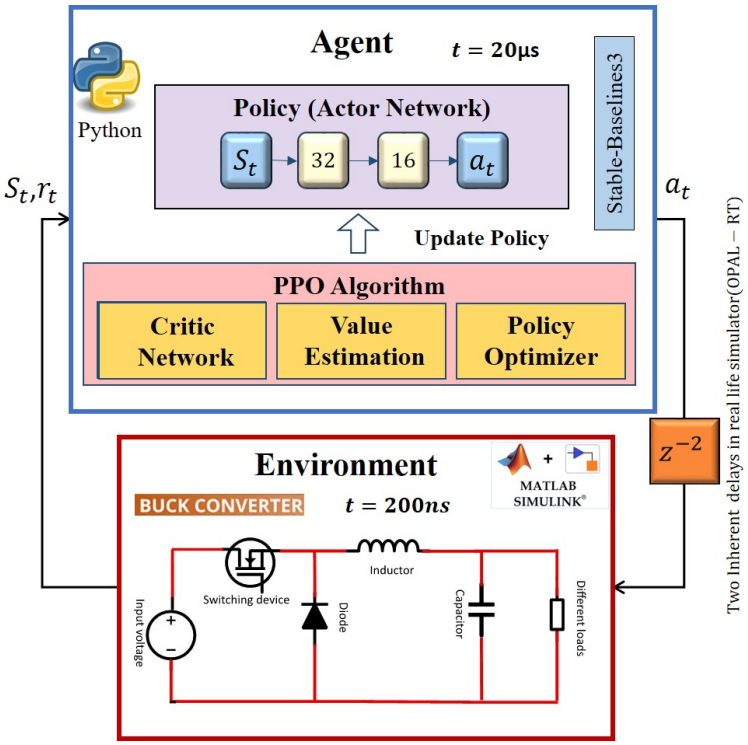


Figure 4. The structure of the PPO agent interacting with the environment.

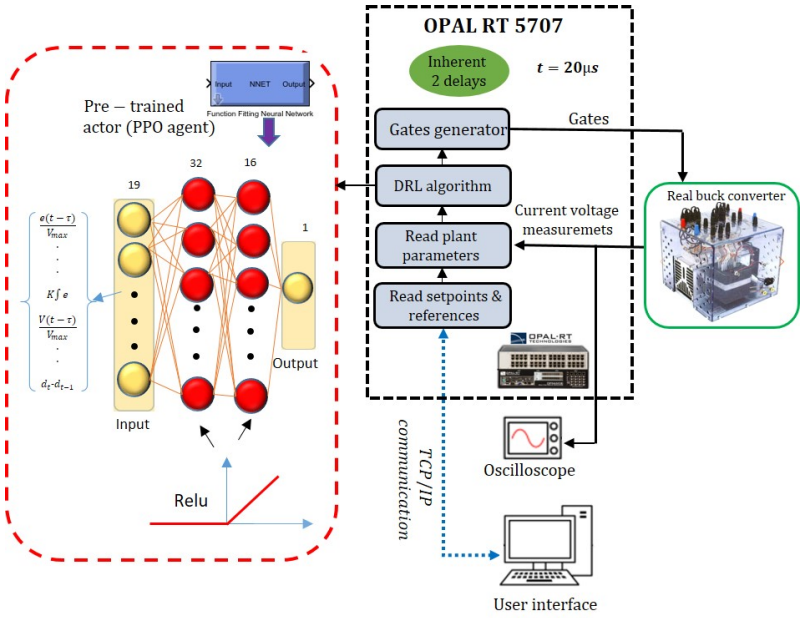


Figure 5. The procedure of implementing the PPO in real life.

Introducing delays in a control system, especially when transitioning from simulation to real-time, can significantly impact performance. To address this, the PPO agent is trained

in a detailed switching model that incorporates fixed delays of two time steps to accurately reflect real-time conditions.

Notably, the switching model includes the IGBT and diode module along with other elements of the buck converter simulation, as well as all considered parasitic elements, as provided in Tables 5 and 6. Moreover, Table 6 represents the parameters of the real buck converter used in the experimental tests, showing that the simulation parameters are similar to the real-time setup parameters.

It is important to note that the trained actor network, a key component of the PPO agent, is integrated and deployed as the controller within Simulink through the genism command in MATLAB.

Table 5. IGBT and diode module parameters.

Parameter	Value
IGBT Module	
Collector–Emitter Voltage, V_{CE0}	0.9 V
On-State Resistance, r_{CE}	30 mΩ
Output Capacitance, C_{oes}	0.2 nF
Diode Module	
Forward Voltage Drop, V_{F0}	1.3 V
Forward Slope Resistance, r_F	20 mΩ

Table 6. The parameter values used in the simulation and experimental setup.

Parameter	Value
Input Capacitor, C_{in}	1100 µF
ESR_{in}	1 mΩ
Output Capacitor, C_{out}	200 µF
ESR_{out}	1.4 mΩ
Inductor, L	250 µH
Parasitic Resistance, r_L	1 mΩ
Resistor load, R_o	$5 < R_o < 80$ Ω
Inductive load, L_o	$20 < L_o < 100$ µH
Input voltage, V_{in}	$150 < V_{in} < 250$ V
Reference voltage, V_{ref}	$50 < V_{ref} < 150$ V
Switching frequency	$20 < f_{sw} < 40$ kHz

4. Simulations and Experimentals Validations

In this part, simulation and implementation tests are applied to a real buck converter to represent the performance of a PPO agent. As it is shown in the following results, the proposed methodology of the PPO performs well in controlling the buck converter.

4.1. Reward Visualization

As can be seen in Figure 6, the training process of PPO is increasing in terms of average episode rewards until it reaches its steady state. In the beginning, the average reward is a large negative value since the DRL agent is initialized with random weights without any knowledge of the control system. During training, the agent learns how to decrease the penalty, although with some oscillations due to parameter variations during training. The number of episodes and the number of steps in each episode are considered 500 and 2000, respectively.

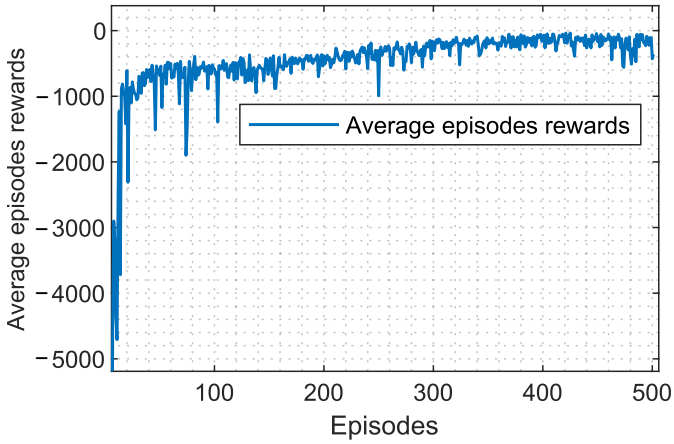


Figure 6. Average accumulated reward over episodes in the PPO training.

4.2. Simulations Analyses and Results

Various scenarios illustrating the performance obtained by using the PPO controller in the buck converter are presented as follows:

4.2.1. Scenario A

This scenario deals with applying perturbations to the input voltage, the references voltage, the output resistive load, as well as variations in the inductive load. Simultaneously, these changes are applied to the system within the exact ranges considered during training. As it is represented in Figures 7 and 8 for this scenario, the proposed agent effectively adjusts its control policy to preserve exact regulations in the buck converter under both resistive and inductive loads and maintains its stability and accuracy during learning in both DCM and CCM. Furthermore, the duty cycle exhibits smooth behavior during variations.

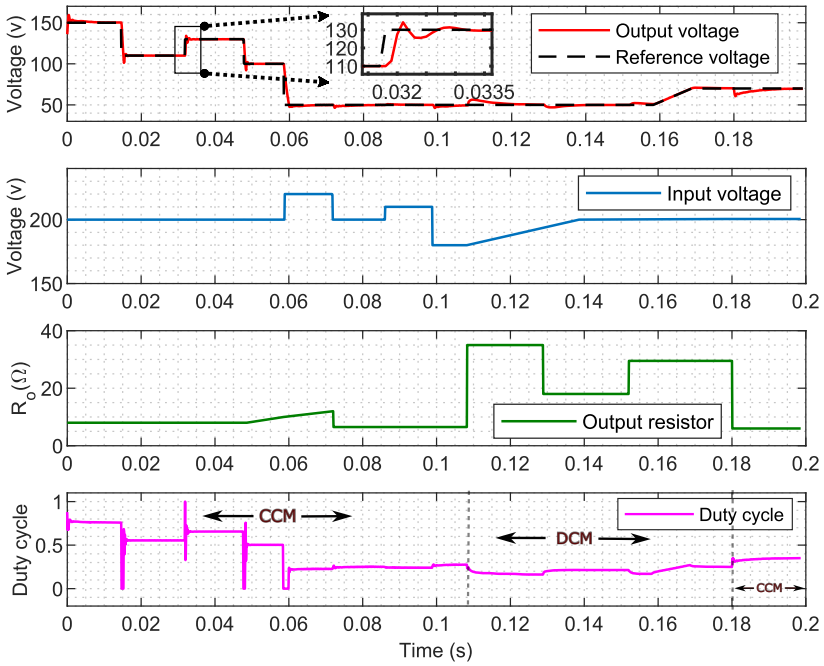


Figure 7. Results of Scenario A during variation inside of the training range for resisitive loads.

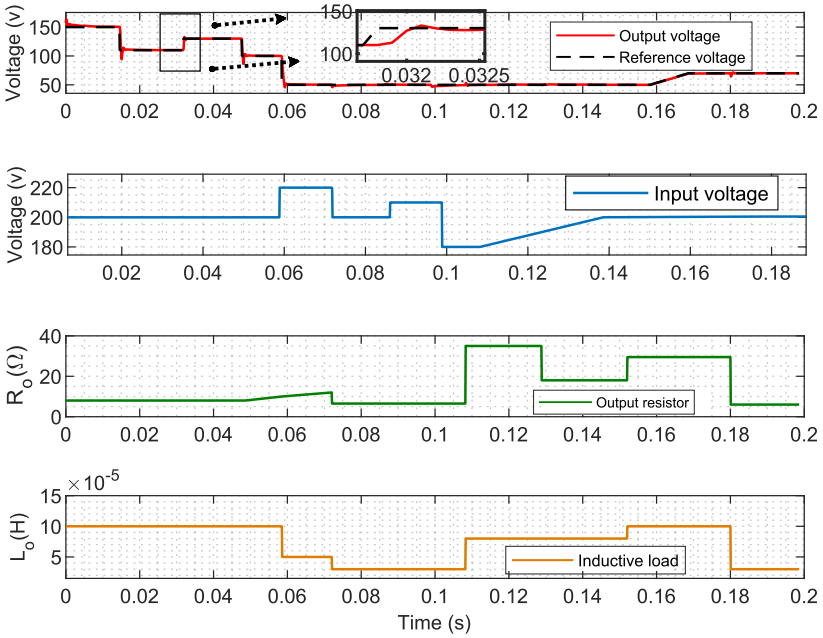


Figure 8. Results of Scenario A during variation inside of the training range for a resistive–inductive load.

4.2.2. Scenario B

This scenario refers to changes in the input voltage and the reference voltage together with output resistor variations occurring outside of its training range. In this scenario, the generalizability of the proposed agent is studied by subjecting the system to variations that extend beyond those seen during training, as depicted in Figure 9. The agent's ability to effectively regulate the buck converter under these out-of-distribution variations highlights its capacity for generalization. This illustrates its competence in handling unforeseen circumstances and extending its applicability to broader operational conditions [43]. In Figure 9, the red dashed lines define the boundary of the training range for the input voltage and the output resistor.

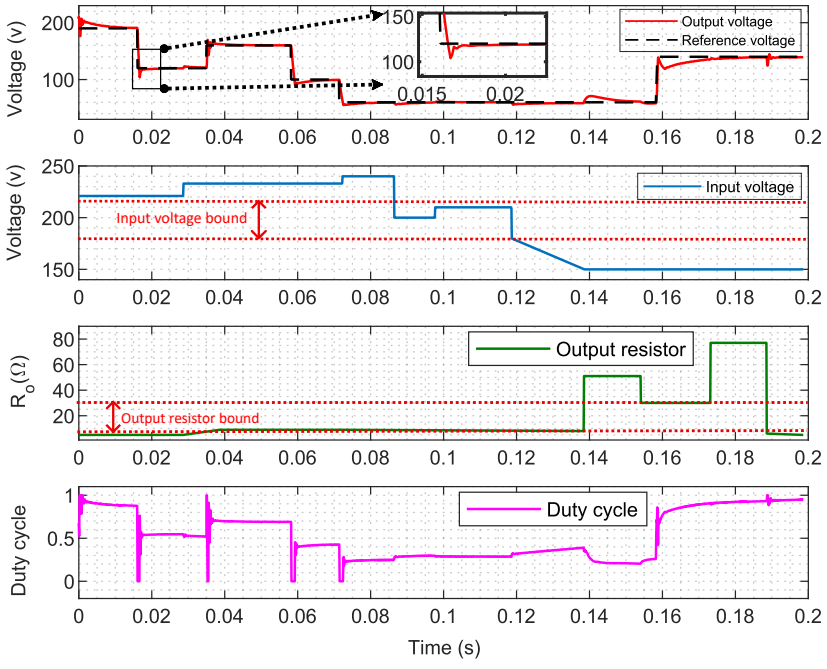


Figure 9. Results of Scenario B during variation outside of the training range for resistive loads.

4.2.3. Scenario C

This scenario indicates the effect of uncertainties on the inductor and capacitor values. The robustness of the PPO in the presence of non-deterministic sources, such as parameter uncertainty, is depicted in Figure 10 to evaluate the system behavior. Parameter variations are applied from the nominal values of the inductor and capacitor as L_n and C_n to their $\pm 20\%$ uncertainties and are shown in Figure 10. The changes in the output resistor and the input voltage are the same as in scenario A in Figure 7.

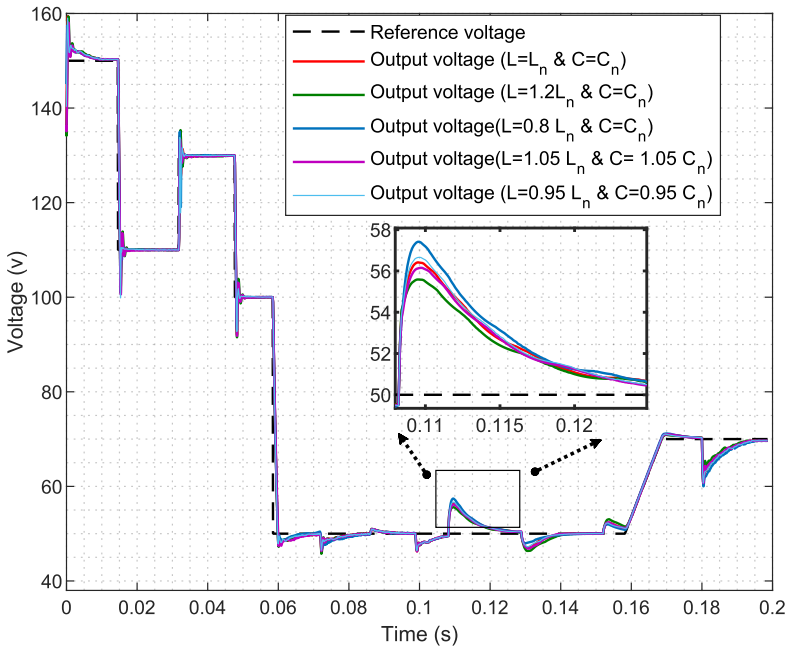


Figure 10. Effect of parameter uncertainty on the output voltage.

4.2.4. Scenario D

The impact of different network sizes is evaluated in this scenario using the PPO algorithm. The simulation results with different numbers of layers and neurons are analyzed in this section for the conditions of Scenario A. Although larger networks (in terms of layers' depth and number of neurons per layer) are more capable of extracting non-linear and complex patterns, they are more prone to overfitting, which reduces the generalizability of the model. In this case, the model may lose its performance with even a small change in the environment. In addition, larger models require more multiply-accumulate operations (MACs) to be performed. This is important since these models are usually deployed on resource-constrained devices. To avoid overfitting and complexity of the model, it is advisable to minimize excessive growth in the size of the model. On the other hand, smaller networks aim to mitigate potential underfitting in the face of the inherent complexity of the control problem. The control performance, measured by the root mean square error (RMSE) for the condition in scenario A, along with memory usage, is depicted in Figure 11 for various selected network sizes, as illustrated in a bar plot. The results illustrate that adjusting the size of the network can lead to improved performance without excessive complexity in executing the actor network. These network arrangements are selected from various reasonable conjectures, with a focus on maintaining non-increasing layer sizes, which are considered best practices. As observed in Figure 11, the structure of $(128, 64, 32)$ yields a small RMSE value but incurs higher computational costs due to increased memory usage. Moreover, through experimental tests, as shown in Figure 12, it is found that the configuration $(16, 8)$, while exhibiting good performance, suffers from chattering issues. Therefore, a structure of $(32, 16)$ is selected to strike a balance between memory usage, computational footprint, and achieving satisfactory performance.

It is worth noting that the feasibility of deploying the proposed network architecture on resource-constrained devices has been evaluated based on computational efficiency, computational complexity, and memory requirements, as detailed in [44–48]. These factors demonstrate the suitability of the proposed approach for easy implementation on cost-effective FPGAs and other resource-constrained platforms, ensuring efficient utilization of hardware resources together with enabling deployment in real-world applications where computational resources are limited. Table 7, presenting the number of MACs alongside different neural network sizes, offers valuable insights into computational complexity, resource requirements, and scalability. It demonstrates that the selected architecture $(32, 16)$ has a relatively low number of MACs, making it suitable for implementation on simple FPGA platforms. While the proposed structure is well-suited for deployment on such devices, the OPAL-RT hardware-in-the-loop platform is chosen for implementation due to its specific features and capabilities.

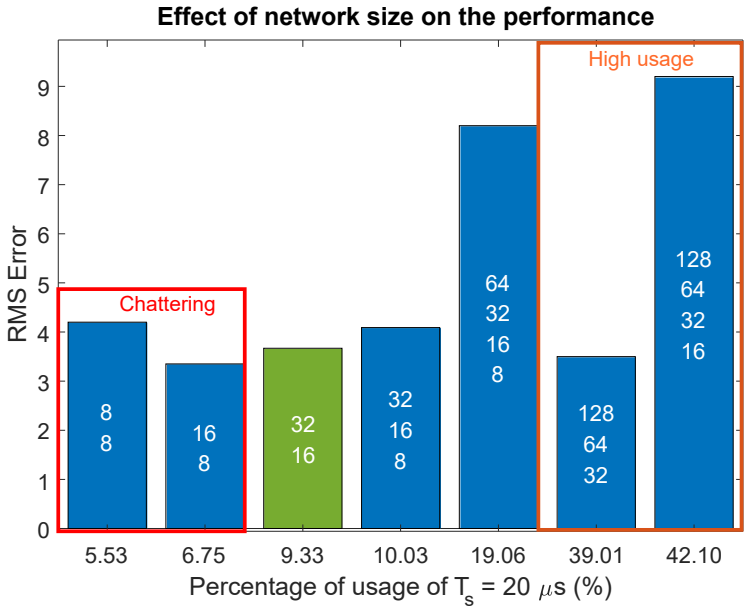


Figure 11. The overall performance in various network sizes.

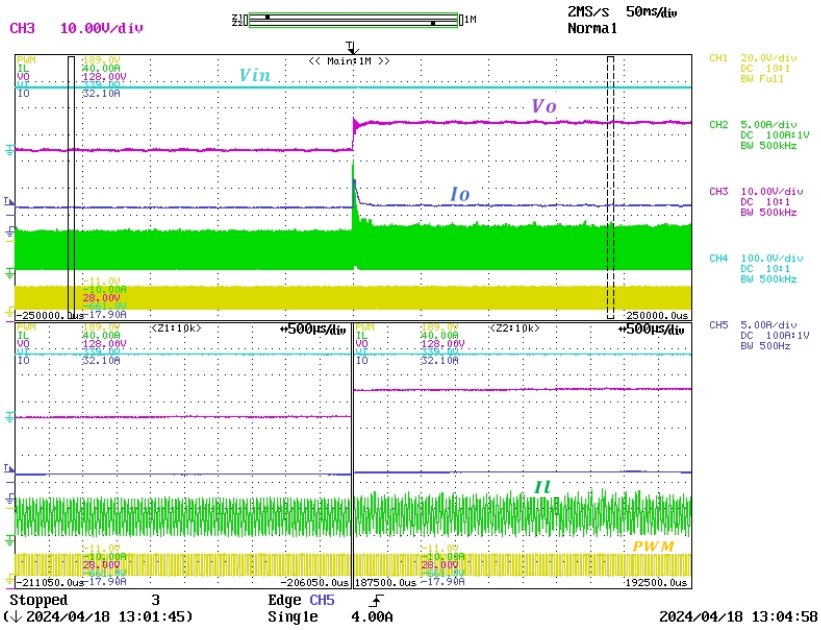


Figure 12. Experimental results for the $(16, 8)$ structure showing chattering.

Table 7. The number of MACs for different network sizes.

Network Size	MACs
(8, 8)	224
(16, 8)	440
(32, 16)	1136
(32, 16, 8)	1256
(128, 64, 32)	12,704
(64, 32, 16, 8)	3912
(128, 64, 32, 16)	13,200

4.2.5. Scenario E

This scenation examines the effect of integral error gain (k), which was previously mentioned in Table 2, on the output response. Figure 13 displays similar variations to those in Scenario A.

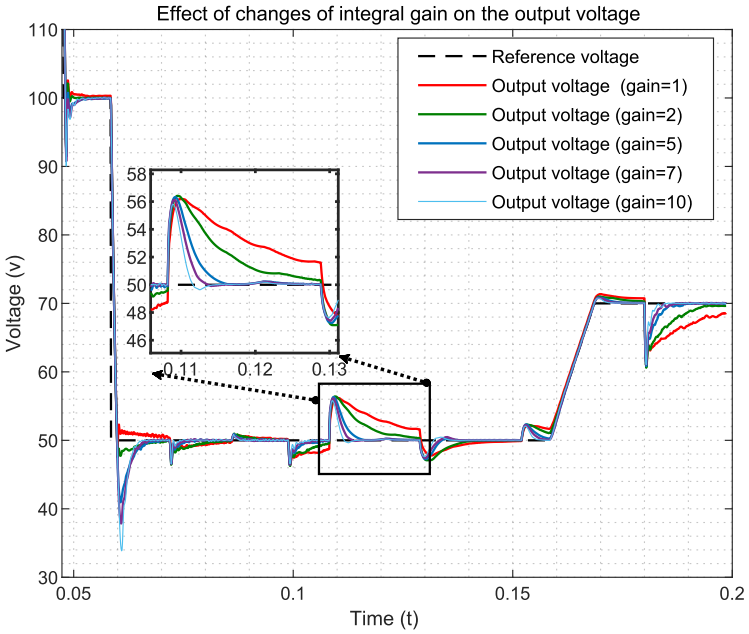


Figure 13. Effect of integral gain on the output response.

It is important to emphasize the significance of modifying the behavior of a transient system without requiring retraining of the model. This flexibility enables quick adaptation to changing system dynamics without the computational overhead of retraining. The output voltage response is displayed for different gain values, revealing distinct transient responses, overshoot characteristics, and settling times. Figure 13 highlights the sensitivity of the system to changes in the integral gain, providing valuable insights into its dynamic behavior under different control configurations. A higher gain accelerates the system's response, potentially achieving faster dynamics. However, this comes at the cost of increased overshoot, which may lead to undesired behavior. Thus, there exists a trade-off between achieving a fast response and limiting overshoot, necessitating careful selection of the gain parameter. In this study, a gain value of 2 is considered, striking a balance between achieving rapid response times while mitigating overshoot to acceptable levels.

4.2.6. Scenario F

This scenario is related to performance validation across different training episodes, which is shown in Figure 14. It is important to consider the trade-off between precision in voltage convergence and the computational time in training. For this reason, after

transferring the pre-trained agent across various episodes in testing, RMSE is computed for each agent at specific episode intervals for the condition in Scenario A.

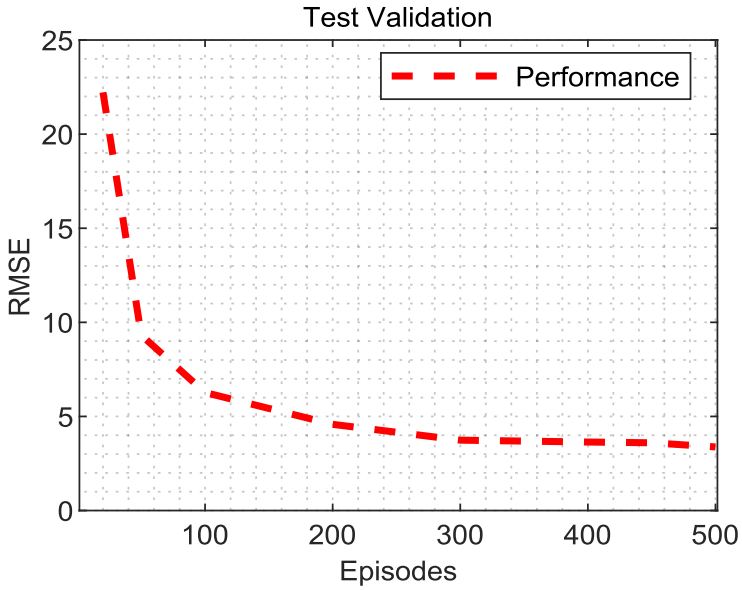


Figure 14. Performance validation in various training episodes.

The observation indicates that a significant enhancement in performance is achieved after approximately 300 training episodes. However, to attain the best performance, a strategic choice is made, and training is extended to 500 episodes.

4.3. Experimental Verifications

To confirm the performance of the proposed control algorithm with the real system and all its challenges, experiments are carried out on the buck converter setup, which is illustrated in Figure 15. The experimental setup includes four principal parts:

(a) The buck converter's parameters are presented in Table 6. In the buck converter setup, the SKM50GB12T4 IGBT module is employed as the power switch and the diode. This module, along with the input capacitor, is integrated within SEMITEACH B6U+E1CIF+B6CI from Semikron Danfoss, Sartrouville, France. Additionally, an inductor and output capacitor are incorporated to assemble the buck converter.
(b) The real-time simulator OPAL-RT-5707, in which the control algorithm is embedded, generates PWM signals (20 kHz), receives the measurements from the sensors, and implements the PPO.
(c) The Cinergia GE&ELvAC/DC versatile converter functions as the electronic load (EL) and the input voltage.
(d) A sensor board is also present to convert measured voltage and current to OPAL compatible format.

The experimental results depicted in Figures 16–19 serve to validate the simulation outcomes for the distinct scenarios, offering real-world evidence of the PPO controller's performance. It is noticeable that the green and yellow lines depict the inductor current and PWM signals, respectively, while the violet line indicates the output voltage. Furthermore, the two blue lines correspond to the input voltage and output current, with light blue representing the input voltage and dark blue representing the output current.

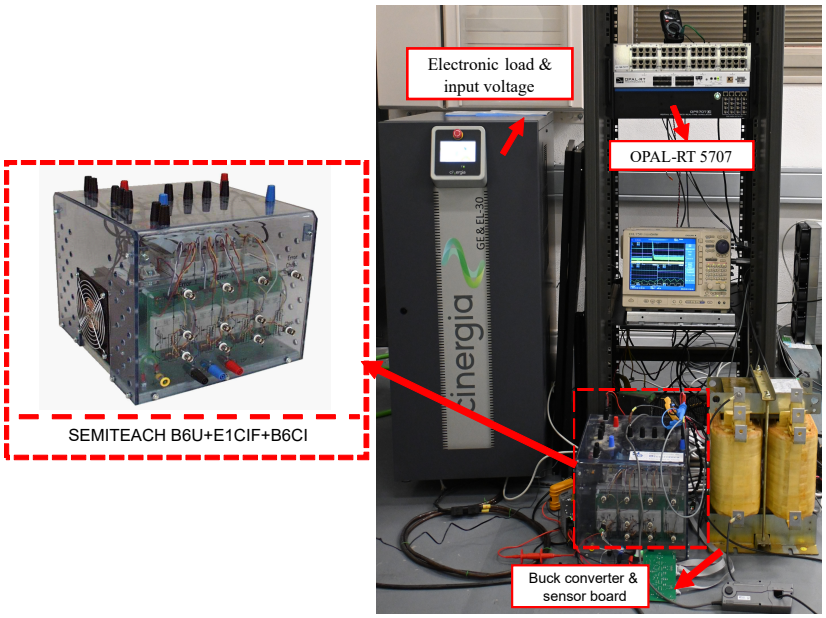


Figure 15. Experimental setup of the buck converter.

4.3.1. Responses under Input Voltage Step Variations

In this test, the reference voltage and the output resistor remain constant, while the input voltage varies from 190 V to 210 V. As depicted in Figure 16, the controller can regulate the output voltage, effectively compensating for fluctuations in the input voltage. The voltage deviation is less than 0.1% in the steady state.

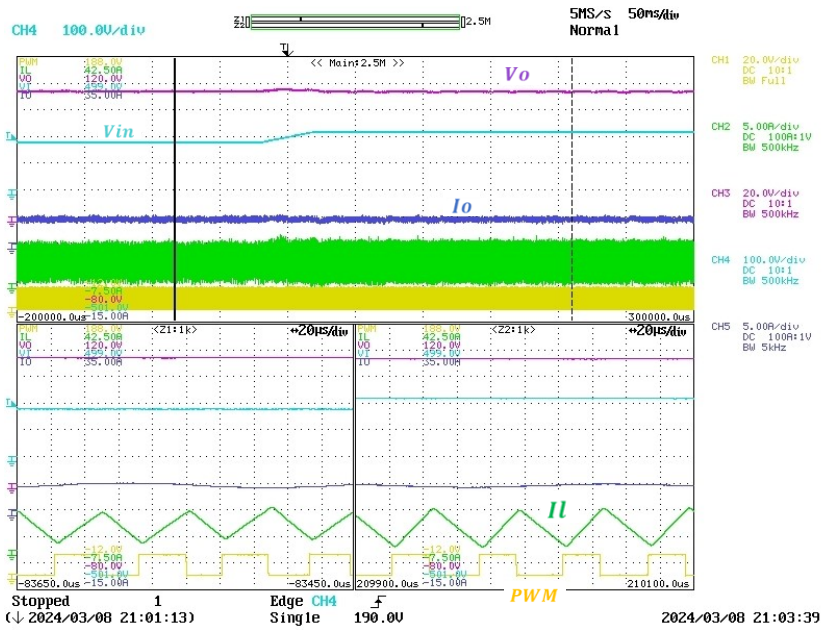


Figure 16. Experimental results of the buck converter with the input voltage step variations.

4.3.2. Responses under Step-Reference Voltage Changes

The input voltage and the output resistor remain unchanged in Figure 17. However, the reference voltage is changed from 50 V to 125 V. It can be observed that the output voltage effectively tracks the reference voltage within a time period of less than 2 ms after the applying changes, resulting in a steady-state error of approximately 0.1%.

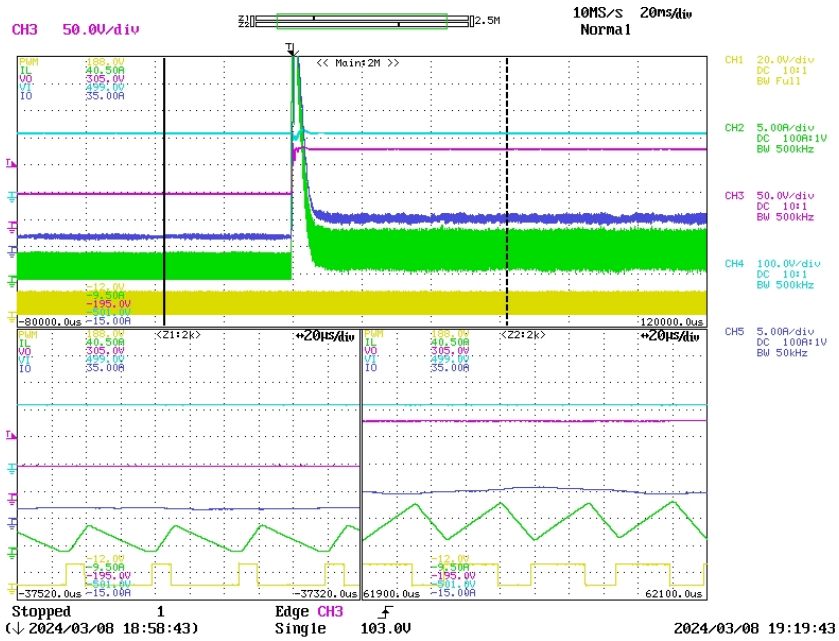


Figure 17. Experimental results for the voltage tracking performance of the buck converter under the variable reference voltage.

4.3.3. Responses under Step Output Resistor Changes

Figure 18 illustrates the regulation of the output voltage in response to the variations in the output resistor from 80 Ω to 10 Ω, while the input voltage and the reference voltage remain constant. It is evident that the buck converter operates appropriately in both CCM and DCM.

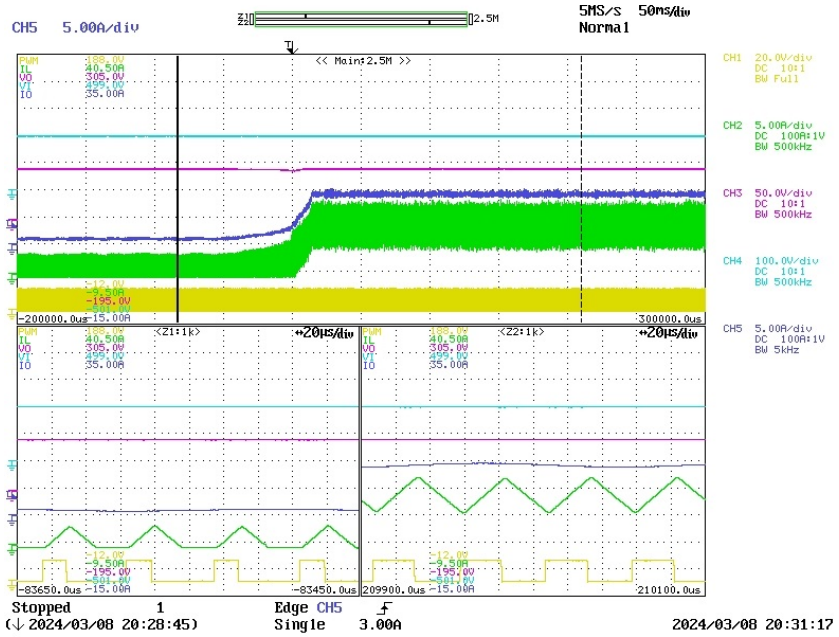


Figure 18. Experimental results of the buck converter with output resistor step changes.

4.3.4. Responses under Switching Frequency Changes

Figure 19 depicts the implementation results of the variations in the switching frequency from 20 kHz to 40 kHz, while other parameters remain constant. The purpose of this test is to evaluate the performance of the PPO agent under varying switching frequencies, demonstrating its robustness in complex conditions. It is evident that the PPO agent can quickly adapt to such changes, whereas traditional controllers may struggle

with frequency variations. Additionally, the buck converter can operate in both DCM and CCM, ensuring smooth transitions between these modes while maintaining stable output voltage regulation.

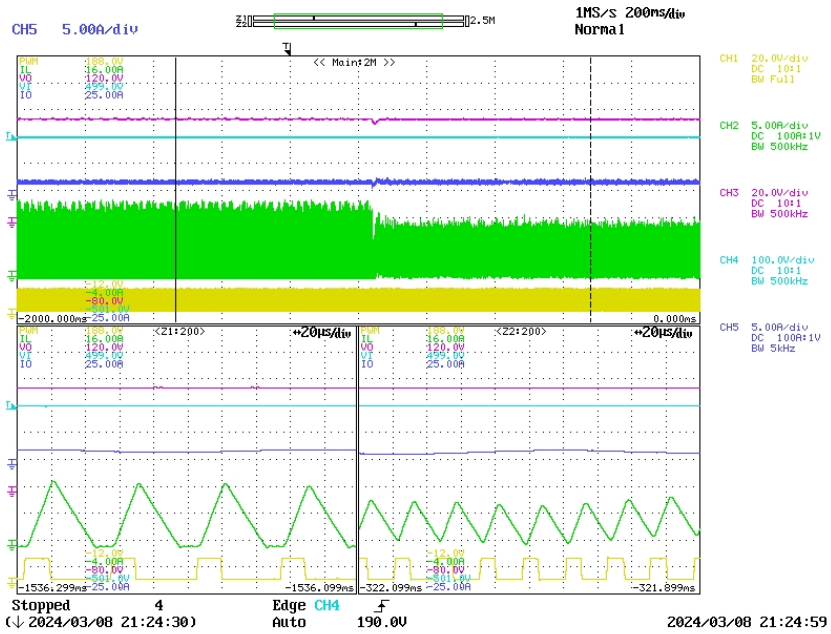


Figure 19. Experimental results of the buck converter with the switching frequency step changes.

5. Conclusions

The adaptability of the PPO controller for the buck converter is enhanced by innovative input features, a chattering reduction technique, and its capacity to operate in both the DCM and CCM, making it highly versatile across different types of loads. Furthermore, this paper emphasizes the agent's dual capabilities: adaptability to variations within the learned range and generalizability to variations beyond the training range. Based on the performed simulation and experimental tests, the proposed approach shows remarkable compatibility with real-time conditions, demonstrating robust performance and stable behavior in the control of a buck converter While maintaining a steady-state error of less than 0.1% and ensuring a short settling time of less than 2 ms in all cases. This makes it a good candidate for achieving the main objectives while addressing uncertainties in different scenarios. This work facilitates the deployment of DRL-based buck converters in more realistic scenarios than previously envisioned. A key consideration in this deployment is the trade-off between memory usage and real-time performance for different neural network sizes. The study highlights how optimizing network size can balance these factors, ensuring efficient real-time control while minimizing computational overhead. The application of the proposed methodology can be extended to control different power electronic converters, such as interleaved buck converters and DAB converters, in future work.

Author Contributions: Methodology, D.P.; Validation, D.S.; Writing—original draft, N.M.; Supervision, E.B. and S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Spanish Ministry of Science, Innovation and Universities, under Grants PID2021-125628OB-C22 and TED2021-130610B-C21. Junta de Comunidades de Castilla La Mancha and the European Union through the European Regional Development Fund: SBPLY/21/180501/000147 Comunidad de Madrid in the framework of the Multiannual Agreement with the Universidad de Alcalá in the action line "Stimulus to the Research of Young Doctors", within the V PRICIT framework program: CM/JIN/2021-019.

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

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

References

1. Becker, D.J.; Sonnenberg, B.J. DC microgrids in buildings and data centers. In Proceedings of the 2011 IEEE 33rd International Telecommunications Energy Conference (INTELEC), Amsterdam, The Netherlands, 9–13 October 2011; pp. 1–7.
2. Bottrell, N.; Prodanovic, M.; Green, T.C. Dynamic Stability of a Microgrid With an Active Load. *IEEE Trans. Power Electron.* **2013**, *28*, 5107–5119. [CrossRef]
3. Hernandez, L.; Baladron, C.; Aguiar, J.M.; Carro, B.; Sanchez-Esguevillas, A.J.; Lloret, J.; Massana, J. A Survey on Electric Power Demand Forecasting: Future Trends in Smart Grids, Microgrids and Smart Buildings. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1460–1495. [CrossRef]
4. Mazaheri, H.; Francés, A.; Asensi, R.; Uceda, J. Nonlinear Stability Analysis of DC-DC Power Electronic Systems by Means of Switching Equivalent Models. *IEEE Access* **2021**, *9*, 98412–98422. [CrossRef]
5. Sumita, R.; Sato, T. PID control method using predicted output voltage for digitally controlled DC/DC converter. In Proceedings of the 2019 1st International Conference on Electrical, Control and Instrumentation Engineering (ICECIE), Batu Pahat, Malaysia, 21–22 November 2019; pp. 1–7.
6. Louassaa, K.; Chouder, A.; Rus-Casas, C. Robust Nonsingular Terminal Sliding Mode Control of a Buck Converter Feeding a Constant Power Load. *Electronics* **2023**, *12*, 728. [CrossRef]
7. Cortes, D.; Alvarez, J.; Alvarez, J. Robust sliding mode control for the boost converter. In Proceedings of the VIII IEEE International Power Electronics Congress, 2002, Technical Proceedings, CIEP 2002, Acapulco, Mexico, 16–19 October 2002; pp. 208–212.
8. Andrés-Martínez, O.; Flores-Tlacuahuac, A.; Ruiz-Martinez, O.F.; Mayo-Maldonado, J.C. Nonlinear Model Predictive Stabilization of DC–DC Boost Converters With Constant Power Loads. *IEEE J. Emerg. Sel. Top. Power Electron.* **2021**, *9*, 822–830. [CrossRef]
9. Prag, K.; Woolway, M.; Celik, T. Data-Driven Model Predictive Control of DC-to-DC Buck-Boost Converter. *IEEE Access* **2021**, *9*, 101902–101915. [CrossRef]
10. Krismer, F.; Kolar, J.W. Closed Form Solution for Minimum Conduction Loss Modulation of DAB Converters. *IEEE Trans. Power Electron.* **2012**, *27*, 174–188. [CrossRef]
11. Saldi, S.; Abbassi, R.; Amor, N.; Chebbi, S. Passivity-Based Direct Power Control of Shunt Active Filter under Distorted Grid Voltage Conditions. *Automatika* **2016**, *57*, 361–371. [CrossRef]
12. Gangula, S.D.; Nizami, T.K.; Udumula, R.R.; Chakravarty, A.; Singh, P. Adaptive neural network control of DC–DC power converter. *Expert Syst. Appl.* **2023**, *229*, 120362. [CrossRef]
13. Saadatmand, S.; Shamsi, P.; Ferdowsi, M. The Voltage Regulation of a Buck Converter Using a Neural Network Predictive Controller. In Proceedings of the 2020 IEEE Texas Power and Energy Conference (TPEC), College Station, TX, USA, 6–7 February 2020; pp. 1–6.
14. Saidi, S.; Chebbi, S.; Jouini, H. Harmonic and reactive power compensations by shunt active filter controlled by adaptive fuzzy logic. *Int. Rev. Model. Simul.* **2011**, *4*, 1487–1492.
15. Saad, S.; Zellouma, L. Fuzzy Logic Controller for Three-Level Shunt Active Filter Compensating Harmonics and Reactive Power. *Electr. Power Syst. Res.* **2009**, *79*, 1337–1341. [CrossRef]
16. Bui, V.-H.; Chang, F.; Su, W.; Wang, M.; Murphey, Y.; Silva, F.; Huang, C.; Xue, L.; Glatt, R. Deep Neural Network-Based Surrogate Model for Optimal Component Sizing of Power Converters Using Deep Reinforcement Learning. *IEEE Access* **2022**, *10*, 78702–78712. [CrossRef]
17. Kishore, P.S.V.; Jayaram, N.; Rajesh, J. Performance Enhancement of Buck Converter Using Reinforcement Learning Control. In Proceedings of the 2022 IEEE Delhi Section Conference (DELCON), Delhi, India, 11–13 February 2022; pp. 1–5.
18. Andalibi, M.; Hajihosseini, M.; Teymoori, S.; Kargar, M.; Gheisarnejad, M. A Time-Varying Deep Reinforcement Model Predictive Control for DC Power Converter Systems. In Proceedings of the 2021 IEEE 12th International Symposium on Power Electronics for Distributed Generation Systems (PEDG), Chicago, IL, USA, 28 June–1 July 2021; pp. 1–6.
19. Zandi, O.; Poshtan, J. Voltage control of DC–DC converters through direct control of power switches using reinforcement learning. *Eng. Appl. Artif. Intell.* **2023**, *120*, 105833. [CrossRef]
20. Hajihosseini, M.; Andalibi, M.; Gheisarnejad, M.; Farsizadeh, H.; Khooban, M.-H. DC/DC Power Converter Control-Based Deep Machine Learning Techniques: Real-Time Implementation. *IEEE Trans. Power Electron.* **2020**, *35*, 9971–9977. [CrossRef]
21. Shi, X.; Chen, N.; Wei, T.; Wu, J.; Xiao, P. A Reinforcement Learning-based Online-training AI Controller for DC-DC Switching Converters. In Proceedings of the 2021 6th International Conference on Integrated Circuits and Microsystems (ICICM), Nanjing, China, 22–24 October 2021; pp. 435–438.
22. Pradeep, D.J.; Noel, M.M.; Arun, N. Nonlinear control of a boost converter using a robust regression based reinforcement learning algorithm. *Eng. Appl. Artif. Intell.* **2016**, *52*, 1–9. [CrossRef]
23. Gheisarnejad, M.; Farsizadeh, H.; Khooban, M.H. A Novel Nonlinear Deep Reinforcement Learning Controller for DC–DC Power Buck Converters. *IEEE Trans. Ind. Electron.* **2021**, *68*, 6849–6858. [CrossRef]

24. Tian, F.; Cobaleda, D.B.; Martinez, W. Deep Reinforcement Learning for DC-DC Converter Parameters Optimization. In Proceedings of the 2022 IEEE 31st International Symposium on Industrial Electronics (ISIE), Anchorage, AK, USA, 1–3 June 2022; pp. 325–330.
25. Qie, T.; Zhang, X.; Xiang, C.; Yu, Y.; Iu, H.H.C.; Fernando, T. A New Robust Integral Reinforcement Learning Based Control Algorithm for Interleaved DC/DC Boost Converter. *IEEE Trans. Ind. Electron.* **2023**, *70*, 3729–3739. [CrossRef]
26. Tang, Y.; Hu, W.; Cao, D.; Hou, N.; Li, Z.; Li, Y.W.; Chen, Z.; Blaabjerg, F. Deep Reinforcement Learning Aided Variable-Frequency Triple-Phase-Shift Control for Dual-Active-Bridge Converter. *IEEE Trans. Ind. Electron.* **2023**, *70*, 10506–10515. [CrossRef]
27. Purohit, C.S.; Manna, S.; Mani, G.; Stonier, A.A. Development of buck power converter circuit with ANN RL algorithm intended for power industry. *Circuit World* **2020**, *47*, 391–399. [CrossRef]
28. Dong, W.; Li, S.; Fu, X.; Li, Z.; Fairbank, M.; Gao, Y. Control of a Buck DC/DC Converter Using Approximate Dynamic Programming and Artificial Neural Networks. *IEEE Trans. Circuits Syst. I Reg. Papers* **2021**, *68*, 1760–1768. [CrossRef]
29. Zengin, S. Reinforcement learning-based control of improved hybrid current modulated dual active bridge AC/DC converter. *Neural Comput. Appl.* **2022**, *34*, 5417–5430. [CrossRef]
30. Sutton, R.S.; Barto, A.G. *Reinforcement Learning*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018.
31. Wang, H.; Ye, Y.; Zhang, J.; Xu, B. A comparative study of 13 deep reinforcement learning based energy management methods for a hybrid electric vehicle. *Energy* **2023**, *266*, 126497. [CrossRef]
32. Larsen, T.N.; Teigen, H.Ø.; Laache, T.; Varagnolo, D.; Rasheed, A. Comparing Deep Reinforcement Learning Algorithms' Ability to Safely Navigate Challenging Waters. *Front. Robot. AI* **2021**, *8*, 738113. [CrossRef] [PubMed]
33. Li, S.E. *Reinforcement Learning for Sequential Decision and Optimal Control*; Springer: Singapore, 2023; ISBN 978-981-19-7783-1.
34. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
35. Govarchinghaleh, Y.A.; Sabaei, M. Dynamic Service Provisioning in Heterogenous Fog Computing Architecture Using Deep Reinforcement Learning. *J. Supercomput.* 2024, *under review*.
36. Biel, D.; Fossas, E. Some experiments on chattering suppression in power converters. In Proceedings of the 2009 IEEE Control Applications, (CCA) & Intelligent Control, (ISIC), St. Petersburg, Russia, 8–10 July 2009; pp. 1523–1528.
37. Si, J.; Harris, S.L.; Yfantis, E. Neural Networks on an FPGA and Hardware-Friendly Activation Functions. *J. Comput. Commun.* **2020**, *8*, 251–277. [CrossRef]
38. Guillod, T.; Papamanolis, P.; Kolar, J.W. Artificial Neural Network (ANN) Based Fast and Accurate Inductor Modeling and Design. *IEEE Open J. Power Electron.* **2020**, *1*, 284–299. [CrossRef]
39. Guimarães, C.J.B.V.; Fernandes, M.A.C. Real-time Neural Networks Implementation Proposal for Microcontrollers. *Electronics* **2020**, *9*, 1597. [CrossRef]
40. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. In Proceedings of the 4th International Conference on Learning Representations (ICLR)—Conference Track, San Juan, Puerto Rico, 2–4 May 2016; pp. 1–10.
41. Zhang, M.; Gómez, P.I.; Xu, Q.; Dragicevic, T. Review of online learning for control and diagnostics of power converters and drives: Algorithms, implementations and applications. *Renew. Sustain. Energy Rev.* **2023**, *186*, 113627. [CrossRef]
42. Górecki, P.; Górecki, K. Methods of Fast Analysis of DC–DC Converters—A Review. *Electronics* **2021**, *10*, 2920. [CrossRef]
43. Packer, C.; Gao, K.; Kos, J.; Krahenbuhl, P.; Koltun, V.; Song, D. Assessing Generalization in Deep Reinforcement Learning. In Proceedings of the ICLR 2019 Conference, New Orleans, LA, USA, 6–9 May 2019.
44. Dhouibi, M.; Ben Salem, A.K.; Saidi, A.; Ben Saoud, S. Accelerating Deep Neural Networks implementation: A survey. *IET Comput. Digit. Tech.* **2021**, *1*, 1–18. [CrossRef]
45. Elshahawy, N.; Wasif, S.A.; Mashaly, M.; Azab, E. A Real-time P-SFA hardware implementation of Deep Neural Networks using FPGA. *Microprocess. Microsyst.* **2024**, *106*, 105037. [CrossRef]
46. Shawahna, A.; Sait, S.M.; El-Maleh, A. FPGA-Based Accelerators of Deep Learning Networks for Learning and Classification: A Review. *IEEE Access* **2019**, *7*, 7823–7859. [CrossRef]
47. Wang, C.; Luo, Z. A Review of the Optimal Design of Neural Networks Based on FPGA. *Appl. Sci.* **2022**, *12*, 10771. [CrossRef]
48. Nguyen, D.-A.; Ho, H.-H.; Bui, D.-H.; Tran, X.-T. An Efficient Hardware Implementation of Artificial Neural Network based on Stochastic Computing. In Proceedings of the 2018 5th NAFOSTED Conference on Information and Computer Science (NICS), Hanoi, Vietnam, 23–24 November 2018; pp. 237–242.

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.