©2025 Published in 11th International Symposium on Intelligent Technologies in Engineering and Science 15-17 November 2025 (ISITES2025 Diyarbakır - Turkey)

PI Sliding Mode Control using DQN for Boost Converter Control

[1,2]*Murat Erhan Çimen
*[1]Faculty of Technology, Department of Electrical and Electronic Engineering Sakarya University of Applied Sciences, Turkey
[2]Biomedical Technologies Application and Research Center (BIYOTAM), Sakarya University of Applied Sciences, Sakarya, Türkiye

Abstract

Nowadays, studies on reinforcement learning are increasing day by day. In this study, DQN algorithm, which is one of the reinforcement learning methods, was used in the control of DC boost converter used especially in power systems, fuel cells, hybrid electric vehicles, and battery technologies. In this direction, PI parameters of boost converter, which was previously controlled by PI Sliding Mode Control, were determined with DQN. As a result, the output voltage of the boost converter was tested with different voltage levels and its performance was evaluated according to the ISE index. As a result, it was seen that the output voltage at the output of the PI Sliding Mode supported DQN controlled boost converter produced better results than the conventional PI controller and followed the reference voltage better.

Key words: Boost Converter, DQN, reinforcement learning, PI, Sliding Mode

1. Introduction

A boost converter is a DC-DC converter widely used in applications such as solar power systems, fuel cells, hybrid electric vehicles, and battery technologies [1], [2]. These converters are used in many applications such as solar power systems, fuel cells, hybrid electric vehicles, and battery technologies. However, controlling the output voltage presents a nonlinear control problem, as it depends on the states of the switching elements, making control strategies challenging.

Proportional-Integral (PI) and Proportional-Integral-Derivative (PID) controllers are commonly used linear control methods due to their simplicity, reliability, and broad applicability [3], [4]. However, these methods often fall short in managing nonlinear systems. For instance, [5] applied PID control to small DC-DC converters for photovoltaic (PV) panels, while [6]utilized PI control with state feedback and Linear Quadratic Regulator (LQR) methods for modeling a boost converter. Additionally, fuzzy logic controllers [7] Model Predictive Control (MPC) [8], sliding mode control [9] have been employed to boost converters to control it.

In recent years, machine learning techniques such as Reinforcement Learning (RL) have emerged as promising methods for controlling nonlinear systems [10], [11]. RL enables an agent to interact with its environment and learn optimal actions to maximize cumulative rewards, making it particularly useful in dynamic environments [12]. In cases involving model uncertainties or

*Corresponding author: Address: Faculty of Technology, Department of Civil Engineering Sakarya University of Applied Sciences, 54187, Sakarya TURKEY. E-mail address: mehmets@subu.edu.tr, Phone: +0264 616 0000-01-02

nonlinear systems, RL has shown advantages over traditional control methods in controlling. However, RL approaches like Q-Learning (QL) face scalability challenges due to the size of the Q-table [11], [13], [14].

The literature highlights the successful application of RL in controlling DC-DC converters. For instance, [15] designed an LQR controller for a buck converter and used RL to optimize its parameters, achieving better results than traditional PID controllers. [16] compared Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Twin Delayed Deep Deterministic Policy Gradient (TD3) for determining PI controller parameters in a boost converter, with TD3 yielding the best performance.

In this study, PI and DQN supported PI Sliding Mode Control was implemented in the control of the boost converter. According to the results obtained, the Boost converter controlled with PI Sliding Mode supported by DQN produced more successful results than the classical PI controller.

2. Materials and Method

The materials and methods used to control the Boost Converter are explained under subheadings.

2.1. Boost Converter

Boost converters are essentially a step-up power converter that takes a low voltage input and provides an output at a much higher voltage. This converter structure consists of a DC voltage source, a diode, an inductor, a capacitor, a load resistor, and a switching element. The general structure of this circuit is shown in Figure 1. In order to control the circuit, the switching element is controlled by converting the input reference signal with a PWM converter. The state space model of the boost converter modeled in [5] is given in Equation 1.

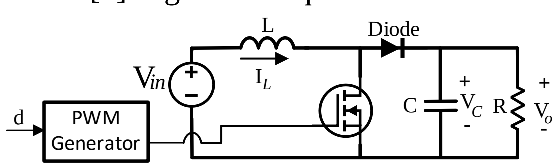

Figure 1. Boost Converter

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \dot{I}_L \\ \dot{V}_C \end{bmatrix} = \begin{bmatrix} 0 & -\dfrac{1-d}{L} \\ \dfrac{1-d}{C} & -\dfrac{1}{RC} \end{bmatrix} x + \begin{bmatrix} \dfrac{1}{L} \\ 0 \end{bmatrix} V_{in} \quad (1)$$

2.2. Sliding Mode Control

One of the powerful methods used in controlling nonlinear systems is sliding mode control. In this method, a surface s is defined first. In order for this surface to come to an equilibrium, it is associated with the Lyapunov equation as in Equation 2. In order for it to come to an equilibrium point over time, it is expected to meet the condition $\dot{V} \leq 0$.

According to the study done in [9] for the sliding mode control of the boost converter, the surface was determined as in equation 3. Then, when the derivative of this surface was taken, the control sign at the balance point was obtained as in equation 4. However, for the condition of being able to slide on the surface, the $-ksign(S)$ expression was added to the control rule and obtained as in equation 5. In this way, the boost converter will remain at the desired reference current value.

$$V = \frac{1}{2}S^2 \rightarrow \dot{V} = S\dot{S} \leq 0 \tag{2}$$

$$S = e = I_L - I_L^* = x_1 - x_1^* \tag{3}$$

$$\dot{S} = \dot{x}_1 - \dot{I}_L^* = \dot{x}_1 - 0 = \frac{1}{L}(V_{in} - (1-d)x_2) = 0 \rightarrow d_{eq} = 1 - \frac{V_{in}}{x_2} \tag{4}$$

$$d = -ksign(S) + d_{eq} \tag{5}$$

2.2. PI Sliding Mode Control

PI controller is used to bring the voltage at the output of the boost converter to the desired level. Equation 6 is given. The error signal entering this controller will create a reference I_L^* current for the boost converter. Since there is an integrator inside the controller, the steady state error will be reset in continuous time. The general diagram of the PI Sliding Mode control structure is given in Figure 2.

$$PI(s) = G_c(s) = \frac{I_L^*(s)}{e(s)} = K_p + K_i\frac{1}{s} \tag{6}$$

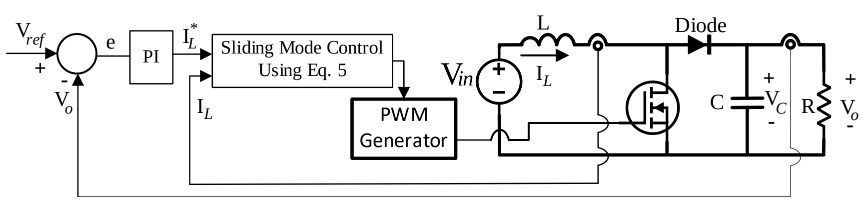

Figure 2. PI Sliding Mode Control of Boost Converter

2.2. Reinforcement Learning

Markov Decision Processes can be used as an expression for problems that can be solved using Reinforcement Learning. In this process, the agent interacts with the environment at each time point and receives a state information and performs an action. After the action is implemented, it receives new state information and a reward from the environment. The general diagram of this structure is given in Figure 3.

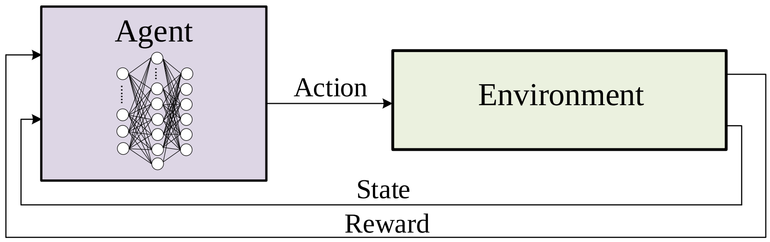

Figure 3. Agent and Environment interaction

DQN algorithm, which can work in a wide state space by combining artificial neural networks with nonlinear problems, has been developed. DQN is a control algorithm that works with temporal difference. This algorithm is updated iteratively as in Equation 7. Here θ constitutes the parameters of artificial neural networks. There are Target and Prediction parts in this equation. The Prediction part is the value that the agent produces according to the situation it is in and the axiom, as given in Equation 8. The Target part expresses the value of the place it will go in the future, as given in Equation 9. The aim is to minimize the cost function in Equation 10. For this, Equation 11 is obtained by taking the derivatives of the equation in Equation 10 according to the artificial intelligence weight parameters θ_i. The weights of the artificial intelligence are updated according to the values obtained from here. Since deep learning is used in this structure, it is called the Deep Q learning algorithm. The pseudocode of the DQN algorithm is given in Algorithm 1.

$$Q(s,a) = Q(s,a) + \alpha \underbrace{\left(\underbrace{r + \gamma \max_{a' \in A} \left(Q(s',a') \right)}_{Target} - \underbrace{Q(s,a|\theta)}_{Prediction} \right)}_{Loss} \tag{7}$$

$$\hat{y} = Q(s,a|\theta) \tag{8}$$

$$\tilde{y} = r + \gamma \max_{a' \in A} \left(Q(s',a') \right) \tag{9}$$

$$L_i(\theta_i) = E_{s,a,r} \left[\left(\tilde{y} - Q(s,a|\theta) \right)^2 \right] \tag{10}$$

$$\begin{aligned} \frac{dL_i(\theta_i)}{d\theta_i} &= \nabla_{\theta_i} L_i(\theta_i) \\ &= E_{s,a,r,s'} \left[r + \gamma \max_{a' \in A} \left(Q(s',a'|\theta_i) \right) - Q(s,a|\theta_i) \right] \nabla_{\theta_i} Q(s,a|\theta_i) \end{aligned} \tag{11}$$

Algorithm 1. Deep Q learning Algorithm

Initialize replay memory D and determine C update step number		
Initialize action values $\tilde{Q}(s,a|\theta)$ function with random weights		
Initialize target value function with specified θ		
For iter=1:M		
	Initialize state $\boldsymbol{s_t}$ in the environment and done=False	
	While done==False	
		Choose $\boldsymbol{a_t}$ action with epsilon greedy
		Execute action $\boldsymbol{a_t}$ to environment and observe state $\boldsymbol{s_{t+1}}$, reward $\boldsymbol{r_t}$, done
		Store transition $D = (\boldsymbol{s_t}, \boldsymbol{a_t}, \boldsymbol{s_{t+1}}, \boldsymbol{r_t})$

		$\hat{y} = \begin{cases} \boldsymbol{r_t} & done = True \\ \boldsymbol{r_t} + \gamma \max\limits_{a' \in A} \left(Q(s', a' \vert \theta_i)\right) & else \end{cases}$
		In order to minimize $L_i(\theta_i) = E_{s,a,r}[(\tilde{y} - \hat{y})^2]$ calculate Eq 11 and update $\tilde{Q}(s, a \vert \theta)$ parameters
		Every C step $\hat{Q} = \tilde{Q}$
	End while	
End For		

2.3. Integration of Reinforcement Learning to PI Sliding Mode Control of Boost Converter

The general framework created for the application of reinforcement learning to the PI Sliding Mode Controlled Boost Converter is given in Figure 4. As seen in Figure 4, action values are applied to the environment. These values are the K_p, K_i values, which are the parameters of the PI controller. According to these values, the Environment produces states and reward values. The generated values are taken by the agent again and the learning process is performed. In addition, the action value is produced according to the relevant situation and applied to the environment again. The states used in this environment are given in equation 12. Actions are calculated in equation 13, the reward value is calculated in equation 14 and the done value is calculated as in equation 15.

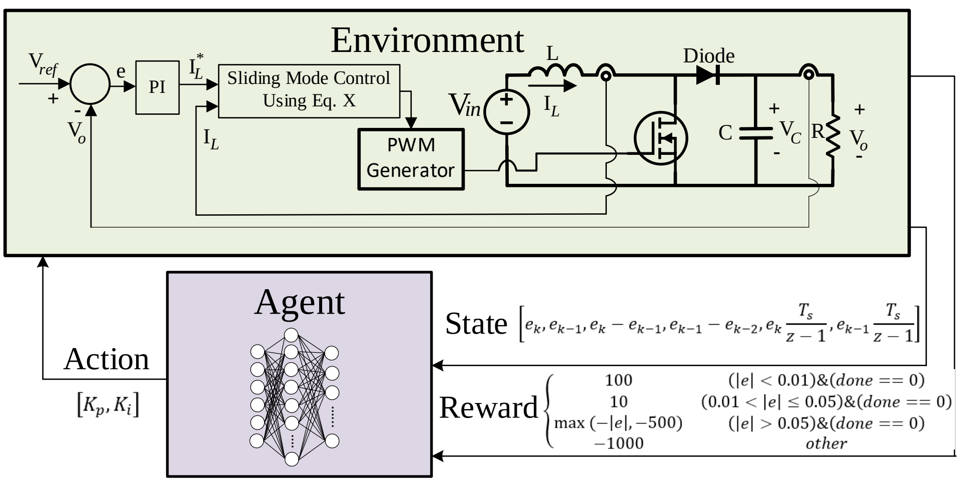

Figure 4. DQN implementation diagram of PI Sliding Mode Control of Boost converter environment

$$State = \left[e_k, e_{k-1}, e_k - e_{k-1}, e_{k-1} - e_{k-2}, e_k \frac{T_s}{z-1}, e_{k-1} \frac{T_s}{z-1}\right] \tag{12}$$

$$Action = \left[K_p, K_i\right] \tag{13}$$

$$Reward = \begin{cases} 100 & (\vert e\vert < 0.01)\&(done == 0) \\ 10 & (0.01 < \vert e\vert \leq 0.05)\&(done == 0) \\ \max(-\vert e\vert, -500) & (\vert e\vert > 0.05)\&(done == 0) \\ -1000 & other \end{cases} \tag{14}$$

$$done = \begin{cases} 1 & Vo > 150 \\ 0 & else \end{cases} \tag{15}$$

In the application, the parameters of the boost converter in the Environment are given in Table 1. The state and action values and ranges designed for the Environment are given in Table 2.

Table 1. Parameters Boost Converter

Component	Symbol	Value	Unit
Resistor	R	50	Ω
Inductor	L	$100x10^{-6}$	Henry
Capacitor	C	$100x10^{-6}$	Farad
Switching Frequency	-	5000	Hz

Table 2. Action Space Ranges of Environment

Action	Kp	0.01, 0.1, 0.5, 1, 1.5, 2
	Ki	0.01, 0.1, 1, 10, 100, 200
State	e_k	-inf, inf
	e_{k-1}	-inf, inf
	$e_k - e_{k-1}$	-inf, inf
	$e_{k-1} - e_{k-2}$	-inf, inf
	$e_k \dfrac{T_s}{z-1}$	-inf, inf
	$e_{k-1} \dfrac{T_s}{z-1}$	-inf, inf

The DQN parameters used in reinforcement learning and the parameter values used for training are given in Table 3.

Table 3. DQN Parameters

Parameter	Parameter Value
Number of Layer	1000
Learning Rate	10^{-3}
GradientThreshold	1
SampleTime	0.01 sec
MiniBatchSize	10^{-4}
EpsilonDecay	10^{-4}
maxepisodes	500
StopTrainingCriteria	EpisodeCount

3. Results

To evaluate the effectiveness of the PI and PI Sliding Mode Control using DQN control algorithm, the converter model is established in Matlab Simulink. Boost Converter with PI Sliding Mode

Control is organized as an environment. The main purpose of the Boost Converter is to ensure that the output voltage value remains at the specified level. This is achieved with both PI and DQN. The results are given in graphs and tables.

PI controller designed by Matlab sisotool tool for the system then it was implemented. Two different controller designs were implemented according to the step response of the model obtained as a result of the design. The first designed controller parameters were determined as Kp=0.003235 Ki=1.7977, while the second controller parameters were determined as Kp=0.007479 and Ki=4.155. Then, the DQN algorithm was arranged to control the environment and the training process was performed. The DQN algorithm applied the action values to the environment. The situations produced by the environment and the reward values were taken and the DQN algorithm was trained. The reward values obtained during the iteration are given in Figure 5.

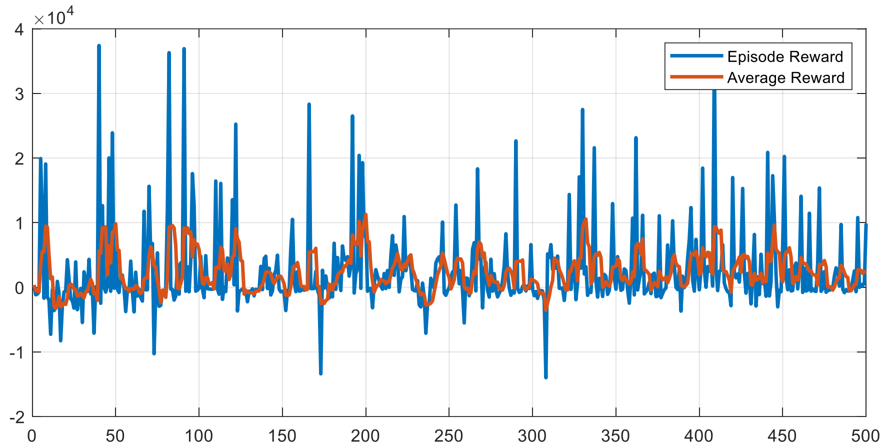

Figure 5. During training, episode reward and average reward values for each iteration

The results of the controlled Boost Converter are given in Figure 6. As can be seen from the results, while the DQN algorithm carries the system to the desired reference value even at different voltage levels, the classical PI controller performed well between certain reference signals but not well with some. Especially at the high reference signal, there were crackles and when it went out of control. The reason for going out of control is that the d signal generated for the PWM exceeds the range of 0-1 limits and the control signal exceeds the limits. The control signals generated for the Boost Converter controlled according to PI and DQN are given in Figure 7.

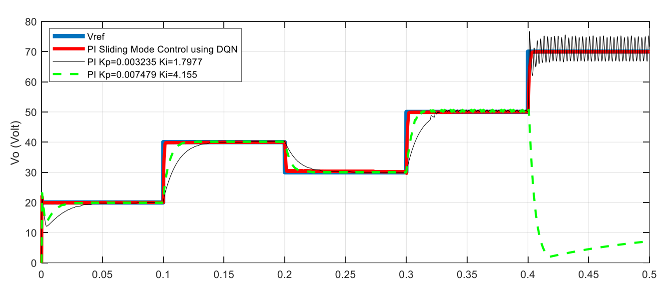

Figure 6. Boost Converter output voltage for PI control and PI Sliding Mode Control using DQN

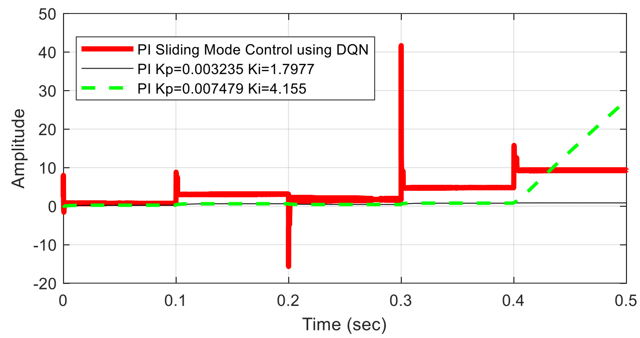

Figure 7. Boost Converter control signal for PI control and PI Sliding Mode Control using DQN

In addition, the control parameters Kp and Ki values produced by DQN are updated at each sampling moment. In return, the system produces a reward value. These values produced by DQN and environment are shown in Figure 8.

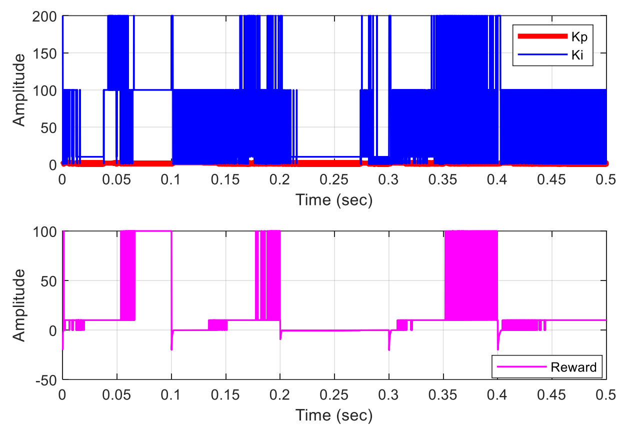

Figure 8. Kp and Ki parameter and reward values for PI Sliding Mode Control using DQN

The performances of PI and PI sliding mode control using DQN methods for the boost converter were compared to the ISE performance values. The results are given in Table 2. As can be seen from the results, the proposed PI Sliding Mode Control using DQN method produced the lowest value and showed much better performance than the classical PI.

Table 2. ISE results of boost converter control

Method	ISE
PI Kp=0.003235 Ki=1.7977	6.5124
PI Kp=0.007479 Ki=4.155	406.2231
PI Sliding Mode using DQN	0.7166

4. Discussion

Boost amplifier, one of the power electronic circuits used to increase DC voltage level, has an important place in many applications. In this study, the output voltage of this amplifier was controlled with conventional PI and PI sliding Mode using DQN and the results were given. When the obtained results were examined, it was seen that the DQN supported PI Sliding Mode controller produced better results in terms of ISE performance index when it controlled the Boost Converter.

Conclusions

In this study, DQN algorithm, which is one of the reinforcement learning methods, was used in controlling the Boost Converter with PI Sliding Mode. Actions, states and reward functions were defined for the PI Sliding Mode controlled Boost Converter designed as an environment. DQN algorithm learned to control the output voltage of the system by updating the parameters of the PI controller at each sampling moment. The results obtained were compared with the classical PI controller. When the results were examined, DQN algorithm showed better performance at different voltage levels and also produced more successful results according to the ISE performance criterion.

References

[1] K. Sundareswaran and V. T. Sreedevi, "Boost converter controller design using queen-bee-assisted GA," *IEEE Transactions on industrial electronics*, vol. 56, no. 3, pp. 778–783, 2008.

[2] M. H. Rashid, *Power Electronics circuits, devices, and applications*. Dorling Kindersley, 2004.

[3] R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, "A Review of PID Control, Tuning Methods and Applications," *Int J Dyn Control*, vol. 9, pp. 818–827, 2021.

[4] M. Çimen, Z. Garip, and A. Boz, "Chaotic flower pollination algorithm based optimal PID controller design for a buck converter," *Analog Integr Circuits Signal Process,* 2021.

[5] O. Güngör and H. İ. Yüksek, "Modeling of Boost and Cuk Converters and Comparison of Their Performance in MPPT," *Sigma Journal of Engineering and Natural Sciences*, vol. 11, no. 1, pp. 83–101, 2020.

[6] M. Alkrunz and İ. Yazıcı, "Design of discrete time controllers for the DC-DC boost converter," *Sakarya University Journal of Science*, vol. 20, no. 1, pp. 75–82, 2016.

[7] A. Sezen and K. Keskin, "Hybrid Control of DC-DC Buck Boost Converter," *Demiryolu Mühendisliği*, vol. 14, pp. 99–109, 2021.

[8] S. Bououden, O. Hazil, S. Filali, and M. Chadli, "Modelling and model predictive control of a DC-DC Boost converter," in *In 2014 15th international conference on sciences and techniques of automatic control and computer engineering (STA)*, 2014, pp. 643–648.

[9] H. Guldemir, “Sliding mode control of DC-DC boost converter,” *Journal of Applied Sciences*, vol. 5, no. 3, pp. 588–592, 2005.

[10] M. E. Harmon and S. S. Harmon, “Reinforcement learning: A tutorial,” *WL/AAFC, WPAFB Ohio*, vol. 45433, pp. 237–285, 1996.

[11] M. E. , Çimen and Z. Garip, “Controlling a Single Tank Liquid Level System with Classical Control Methods and Reinforcement Learning Methods,” *Kocaeli Journal of Science and Engineering*, vol. 7, no. 1, pp. 30–41, 2024.

[12] A. Angiuli, J. P. Fouque, and M. Laurière, “Unified reinforcement Q-learning for mean field game and control problems,” *Mathematics of Control, Signals, and Systems*, vol. 34, no. 2, pp. 217–271, 2022.

[13] W. You, G. Yang, J. Chu, and C. Ju, “Deep reinforcement learning-based proportional–integral control for dual-active-bridge converter,” *Neural Comput Appl*, vol. 35, no. 24, pp. 17953–17966, 2023.

[14] M. E. Çimen, Z. Garip, Y. Yalçın, M. Kutlu, and A. F. Boz, “Self Adaptive Methods for Learning Rate Parameter of Q-Learning Algorithm,” *Journal of Intelligent Systems: Theory and Applications*, vol. 6, no. 2, pp. 191–198, 2023.

[15] D. Alfred, D. Czarkowski, and J. Teng, “Reinforcement Learning-Based Control of a Power Electronic Converter,” *Mathematics*, vol. 12, no. 5, p. 671, 2024.

[16] R. F. Muktiadji, M. A. Ramli, and A. H. Milyani, “Twin-Delayed Deep Deterministic Policy Gradient Algorithm to Control a Boost Converter in a DC Microgrid,” *Electronics (Basel)*, vol. 13, no. 2, p. 433, 2024.