DOI: https://doi.org/10.62985/j.huit_ojs.vol26.no2E.376

A DIGITAL IMPLEMENTATION OF ADAPTIVE PI CONTROLLER FOR DC–DC BOOST CONVERTER BASED ON REINFORCEMENT LEARNING

Vi-Do Tran*, Anh Khoi Tran

HCMC University of Technology and Engineering,

No 1 Vo Van Ngan Street, Thu Duc Ward, Ho Chi Minh City, Vietnam

*Email: *dotv@hcmute.edu.vn*

Received: 23 January 2026; Revised: 3 February 2026; Accepted: 17 April 2026

ABSTRACT

DC–DC boost converters are widely applied in various industrial power electronics applications, such as motor drive systems and voltage step-up stages in grid-connected renewable energy systems, due to their simple and compact circuit topology combined with high voltage gain. However, because of the inherent right-half-plane zero in their dynamic characteristics, feedback control of DC–DC boost converters is regarded as one of the most challenging nonlinear control problems. To date, numerous efforts have been devoted to improving the performance of boost converters. Among modern approaches, reinforcement learning has attracted significant attention for its potential in designing adaptive controllers through the automatic tuning of conventional control structures, including PID, PI, and LQR controllers. This paper proposes an implementation pipeline for a DC–DC boost converter based on an ESP32 microcontroller platform and a 16-bit ADS1115 ADC for constructing the training environment of an adaptive RL-PI controller. In addition, control performance evaluation indices are established based on standard criteria, including steady-state error, overshoot, and output voltage ripple, enabling a quantitative and objective comparison among different control strategies. The controllers investigated in this study include a conventional PI controller, a cascaded PI controller with voltage and inductor current feedback, and the proposed adaptive RL-PI controller, all evaluated under varying load conditions. Experimental results demonstrate that the reinforcement learning–based approach exhibits strong potential for adaptive control of boost converters, achieving output voltage ripple below 2% and steady-state error below 5%.

Keywords: Adaptive controller, boost DC-DC converter, reinforcement learning.

1. INTRODUCTION

In recent years, the rapid advancement of the semiconductor industry has significantly improved power semiconductor devices, not only in terms of their ability to withstand higher current and voltage levels but also in reducing power losses. In parallel, large-scale manufacturing technologies have contributed to a continuous reduction in the cost of power electronic components. These prerequisites have strongly promoted the development of DC power converters, including the boost DC–DC converter, which converts a DC input voltage to a higher DC output voltage, has been widely applied in industrial power systems and renewable energy applications due to its compact structure and high conversion efficiency. However, the mathematical model of the boost DC–DC converter inherently exhibits a right-half-plane zero, which renders the system non-minimum phase. Consequently, the boost converter is regarded as one of the most challenging nonlinear control systems to stabilize, posing significant

difficulties for researchers [1], [2]. Numerous control strategies for boost converters have been investigated, spanning multiple levels of complexity depending on the degree of system modeling. Among linear control-based approaches, the work reported in [3] introduced a fractional-order PID controller designed using inductance and capacitance parameters, demonstrating the feasibility of enhancing conventional PID controllers. Experimental results confirmed improved performance of the DC–DC converter. To further improve control accuracy when employing PID controllers for boost converters, the resolution of measurement signals has also been studied. The authors in [4] demonstrated that a hardware implementation using microcontrollers and 10-bit ADC/DAC converters significantly improved system performance, achieving a voltage ripple of 0.8% and a conversion efficiency of up to 95%. In [5], a digital PID controller tuned using the Ziegler–Nichols (Z–N) method was implemented on an Arduino Mega platform for a boost DC–DC converter, where both simulation and experimental results indicated satisfactory voltage-tracking performance. Nevertheless, under output load variations such as short-circuit conditions, which may cause voltage sag, or open-circuit conditions, which may lead to overvoltage, conventional linear PID controllers often struggle to maintain long-term system stability. To address these challenges, more nonlinear control schemes have been developed, particularly automatic parameter-tuning algorithms for PID controllers. In addition to conventional approaches, modern intelligent control methods have also been proposed, including black-box-based decision-making systems that replace fixed parameters in linear controllers such as PID and LQR [2]. These include machine learning–based optimization methods such as genetic algorithms (GA), artificial neural networks (ANN), and swarm optimization approaches, including PSO-based intelligent PID controllers, as reported in [6]-[9]. In [10], the authors proposed an ANN–BF architecture for tuning PID parameters to control a four-switch boost converter, and simulation results highlighted the potential of intelligent adaptive controllers for high-performance buck–boost systems. Furthermore, [11] presented an adaptive PID controller based on backpropagation neural networks with an optimized dual learning-rate design, which significantly reduced overshoot and transient response time in an H-bridge switched-mode power supply, as demonstrated through simulation. However, the implementation of adaptive PID controllers based on machine learning techniques often faces challenges related to data availability, since it is difficult to capture all possible load and input voltage variations. Additionally, the complexity of real-time training and hardware constraints, such as avoiding prolonged saturation of semiconductor switching devices during training convergence, further complicate practical deployment. The emergence of reinforcement learning (RL) and deep reinforcement learning in recent years has inspired extensive research interest. By enabling agents to interact with the environment, RL-based controllers can automatically adapt and tune control parameters through trial-and-error learning without prior knowledge of optimal system parameters. This characteristic makes RL particularly suitable for black-box controllers such as PID [1]-[14]. In [12], an adaptive control scheme for wind turbine speed regulation was successfully investigated by optimizing a fuzzy controller using reinforcement learning. The authors developed an ANFIS–RL model that integrates an agent with an ANN-based architecture for learning evaluation, achieving stable system performance with no overshoot and fast response time. Q-learning is one of the most representative reinforcement learning algorithms, in which actions are updated based on reward signals and learning-rate adjustments. With the rapid development of deep learning and artificial neural networks, new opportunities have emerged to enhance Q-learning agents [15]. In [16], a double Deep Q-Network (DQN) architecture was proposed to improve Q-learning-based control by integrating artificial neural layers into the agent training loop. Simulation results demonstrated the effectiveness of RL-based adaptive controllers for buck converters under the challenges posed by constant power loads (CPLs) in DC microgrid operations. Overall, simulation results reported in previous studies have demonstrated the strong potential of reinforcement learning techniques for

developing adaptive PID controllers. However, the hardware implementation of RL-based controllers such as RL-PID for boost DC–DC converters still faces significant challenges, including implementation cost and component lifespan. In this paper, we propose a comprehensive framework for developing an adaptive RL-PI controller that automatically tunes proportional and integral gains with the objective of improving output voltage response and efficiency under load variations. The proposed framework is divided into two main components. First, an environment is constructed to evaluate output quality under varying load conditions based on voltage ripple and steady-state error. Voltage feedback calculations are performed using an ESP32 microcontroller in conjunction with a 16-bit ADS1115 ADC. Second, a Q-learning–based agent is designed to learn optimal adjustments of the proportional gain K_p and integral gain K_i in response to real-time system disturbances. The training process, consisting of 500 episodes, is conducted entirely online on the hardware platform and demonstrates reliable agent convergence under real-world disturbances with low environmental feedback latency. This research does not aim to propose a new theoretical advance for the Q-learning algorithm; instead, we focus on bridging the gap between reinforcement learning simulation and practical hardware implementation. The main contribution of this work is the development of a practical and cost-effective hardware framework for training an adaptive PI controller based on a Q-learning algorithm directly on a boosted DC-DC converter to efficiently handle real-world measurement disturbances and noise has been proposed. The proposed framework serves as a practical reference for validating the feasibility of directly training reinforcement learning models on power semiconductor hardware. It enables systematic performance evaluation of the DC–DC converter based on steady-state error, voltage ripple, and overshoot, thereby facilitating the implementation of similar reinforcement learning–based control strategies in future studies.

2. MATHEMATICAL MODEL OF BOOST CONVERTER

The schematic of a basic DC–DC boost converter is illustrated in Figure 1, in which the main power switch Q and the inductor L are the key components determining the voltage step-up capability of the circuit. In the experimental setup, Q is implemented using a MOSFET switch, while D denotes a Schottky diode. The converter is supplied by a DC input voltage of E=12 V. The inductance and output capacitance are selected as L=100 μH and C=1100 μF, respectively, to achieve continuous conduction mode operation and reduce output voltage ripple. The nominal load resistance is set to R=470 Ω. The frequency of PWM signal is 50 kHz. The operating principle of the boost DC–DC converter can be described over one PWM switching period, which consists of two distinct intervals: the switch-on interval T_{on} and the switch-off interval T_{off}.

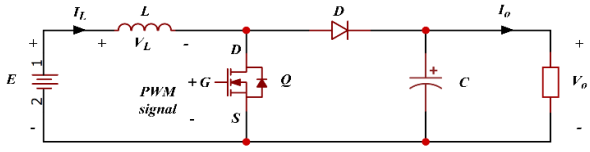

Fig. 1. Basic schematic of the boost converter

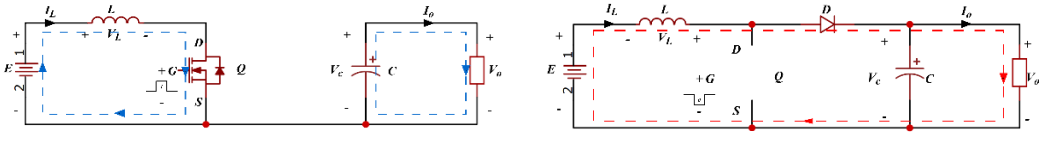

Fig. 2. A) Switch-on interval, B) Switch-off interval

During the switch-on interval illustrated in Figure 2A, when the switch is turned on, the MOSFET drain is driven into saturation and is therefore approximately shorted to the source terminal. As a result, the input voltage E is applied across the inductor L, allowing energy to be stored in the inductor. The differential equations describing the output voltage $v_o(t)$, and the inductor current $i_L(t)$ can be derived from Kirchhoff's Voltage Law (KVL) as follows:

$$\{\dot{i_L}(t) = \frac{E(t)}{L}\ \dot{v_o}(t) = -\frac{v_o(t)}{RC} \tag{1}$$

The state-space equations are formulated from the differential equation set (1):

$$A_1 = \left[0\ 0\ 0\ -\frac{1}{RC}\right];\ B_1 = \left[\frac{1}{L}\ 0\right];\ C_1 = [0\ 1\ ];\ D_1 = 0$$

During the switch-off interval $T_{off}(t)$, as illustrated in Figure 2B, the drain–source path of switch Q is approximately open-circuited. Due to the abrupt change in voltage, the inductor current in L generates an induced electromotive force that appears in series with the input source E. This condition forward-biases the diode, allowing energy to be transferred to the output capacitor C. Consequently, the output voltage $v_o(t)$ is increased by the inductor voltage $v_L(t)$, resulting in an output voltage higher than the input voltage E, thereby achieving the voltage step-up operation of the boost converter. The differential equations describing the switch-off interval are given as follows:

$$\{\dot{i_L}(t) = -\frac{v_o(t)}{L} + \frac{E(t)}{L}\ \dot{v_o}(t) = \frac{i_L(t)}{C} - \frac{v_o(t)}{RC} \tag{2}$$

The state-space equations are formulated from the differential equation set (2):

$$A_1 = \left[0\ -\frac{1}{L}\ \frac{1}{C}\ -\frac{1}{RC}\right];\ B_1 = \left[\frac{1}{L}\ 0\right];\ C_1 = [0\ 1\ ];\ D = 0$$

In this study, small-signal analysis is employed to linearize and approximate equation (2) and (4) according to the following relationship:

$$\{A = A_1 d(t) + A_2[1 - d(t)]\ B = B_1 d(t) + B_2[1 - d(t)]\ C = [0\ 1\ ] \tag{3}$$

Where A, B and C are the state-space matrices obtained by averaging the corresponding matrices A_1, B_1, A_2, B_2. The term $d(t)$ denotes the duty cycle, which is time-varying and determined by the controller. The resulting averaged state-space model is expressed as follows:

$$A = \left[0\ -\frac{1 - d(t)}{L}\ \frac{1 - d(t)}{C}\ -\frac{1}{RC}\right]; B = \left[\frac{1}{L}\ 0\right];\ C = [0\ 1\ ];\ D = 0 \tag{4}$$

The signals $i_L(t);\ v_o(t);\ E(t)$ and $d(t)$ are considered in the small-signal domain and are decomposed into steady-state components and small perturbations δ as follows:

$$\{i_L(t) = I_L + \delta i_L(t)\ v_o(t) = V_o + \delta v_o(t)\ E(t) = E = const\ d(t) = D + \delta d(t)$$

Substituting $i_L(t), v_o(t), E(t), d(t)$ into the averaged state-space equation (4), and rearranging the resulting expressions into steady-state terms, first-order perturbation terms, and second-order perturbation terms, the following relationship is obtained:

$$A = \left[0\ \frac{(D-1)}{L}\ \frac{(1-D)}{C}\ -\frac{1}{RC}\right];\ B = \left[\frac{V_o}{L}\ -\frac{I_L}{C}\right];\ C = [0\ 1\ ];\ D = 0 \tag{5}$$

The steady-state quantities V_o and I_L are determined by the following expressions:

$$I_L = \frac{E}{(1-D)^2 R} \tag{6}$$

$$V_o = \frac{E}{1-D} \tag{7}$$

Based on the derived quantities and system matrices in Eqs. (5),(6) and (7), the open-loop transfer function of the boost DC–DC converter is obtained as follows:

$$G(s) = \frac{\delta V_o(s)}{\delta D(s)} = \frac{\left[\frac{-E}{RC(D-1)^2}\right]s + \frac{E}{LC}}{s^2 + \frac{1}{RC}s + \frac{(D-1)^2}{LC}} \tag{8}$$

From the magnitude and phase Bode plots of the transfer function in (8) is illustrated in Figure 3, evaluated at D=0.5, it can be observed that the open-loop system exhibits a negative gain margin of GM=−33.6 dB at a frequency of 870 rad/s, along with a negative phase margin of PM=−1.23. These characteristics indicate that the system is inherently unstable.

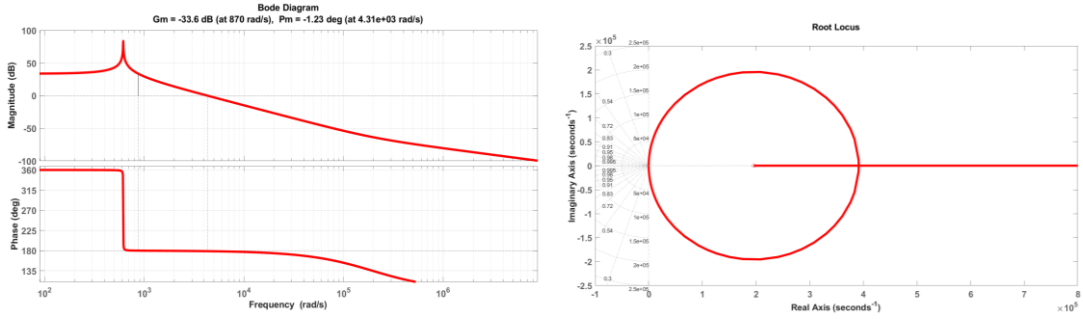

Fig. 3. Magnitude and phase Bode plots and Root locus of the system

The root locus of the system with unity voltage feedback confirms the observations obtained from the Bode analysis. Notably, due to the numerator of the open-loop transfer function in (8), the system inherently exhibits a right-half-plane zero. As a result, only a limited region of the root locus corresponds to stable closed-loop operation.

3. DESIGN ADAPTIVE PI CONTROLLER

3.1. Proportional – Integral (PI) controller

A basic PID controller is described by Eq. (9), consisting of three terms for regulating the voltage feedback error: the proportional, integral, and derivative components. The corresponding tuning parameters are the proportional gain K_p, the integral gain K_i, and the derivative gain K_d. The proportional gain K_p directly influences the steady-state error and the transient response of the system. The inclusion of an ideal integral term significantly reduces the steady-state error, potentially driving it close to zero; however, an increased K_i may also lead to a noticeable rise in system overshoot. The derivative gain K_d plays an important role in improving transient response and reducing overshoot. Nevertheless, transient performance is not the primary design objective for boost DC–DC converters. Instead, steady-state error and voltage ripple are the dominant performance criteria. Therefore, this study focuses on a proportional–integral (PI) controller for the design of the adaptive reinforcement learning–based control scheme.

$$G_c(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{d(e(t))}{dt} \tag{9}$$

3.2. Q-learning algorithm for boost DC-DC converter

Reinforcement learning techniques, including the Q-learning algorithm is illustrated in Figure 4, are machine learning approaches inspired by conditioned reflexes developed through

long-term human training processes. Unlike data-driven machine learning algorithms that rely heavily on object features, such as Artificial Neural Networks (ANNs) or Convolutional Neural Networks (CNNs), reinforcement learning focuses on the interaction between an initialized agent and a training environment to acquire behavioral responses.

The Markov Decision Process (MDP) is a general reinforcement learning framework applied to most RL algorithms and is characterized by the parameter set *(s,a,R,P,γ)*. Within the MDP framework, the RL training process enables the agent, at a given state *s*, to select an action *a* at each time step. Subsequently, the environment evaluates the resulting next state *s′* caused by action *a* and provides a corresponding reward *R(s,a)*. The objective of the training process is to construct an action policy *π(s,a)* that maximizes the expected cumulative discounted reward in the future. This quantity is defined as the return [16], [17]:

$$G = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \tag{10}$$

Where $\boldsymbol{r_{t+k}}$ denotes the instantaneous reward at time step $t+k$. Based on the Markov property, the Q-learning algorithm aims to learn the policy state–action value function $Q^{\pi}(s,a)$, which represents the expected return G when the agent executes action a in state s while following the optimal policy. The Q-value is iteratively updated according to the Bellman equation:

$$(s,a) = (1-\alpha)Q(s,a) + \alpha\left[\gamma Max\left(Q(s',a')\right) + R(s,a)\right] \tag{11}$$

Where $\boldsymbol{s'}$ and $\boldsymbol{a'}$ denote the subsequent state and action of the agent after evaluating system performance at each update of the PI controller parameters. The experience sharing coefficient α reflects the influence of prior and subsequent agent experiences, while the discount factor γ is tuned to enhance the learning speed and convergence performance of the model.

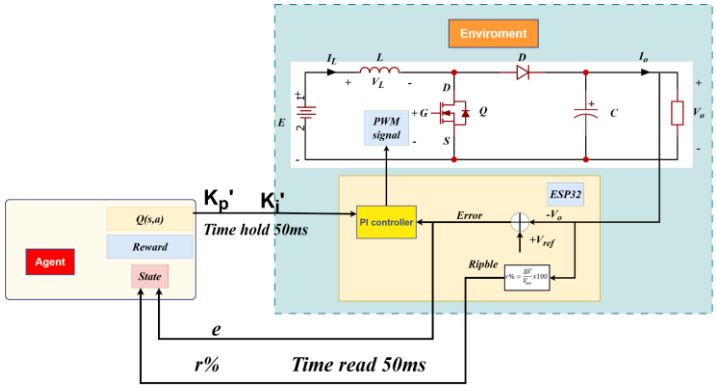

Fig. 4. An adaptive proportional–integral (PI) controller is proposed

A single training step for updating the Q-value *Q(s,a)* based on the Bellman equation is carried out through two sequential phases. Initially, the agent is initialized within a Python-based environment and continuously acquires ripple magnitude and steady-state error values computed by the ESP32 microcontroller. Subsequently, the model determines the corresponding reward or penalty *R(s,a)* based on ripple and steady-state error as well as the current circuit state *s* is illustrated in Table 1. These operations are performed during the state acquisition interval, referred to as the *time read* period (5 ms). Following the Markov decision process, the model selects actions from a predefined action set, including increasing the proportional gain K_p, decreasing K_p, increasing the integral gain K_i, and decreasing K_i. This action set is represented by the action vector:

$$A = (a_1, a_2, a_4, a_5) = \left(+K_p, -K_p, +K_i, -K_i\right) \tag{12}$$

$$a_i \epsilon Q^{\pi}(s, a) \tag{13}$$

After transmitting the updated controller parameters K_p' and K_i' according to the selected action vector, an additional interval is required for these parameters to take effect on the boost converter system. This interval is defined as the *time hold* period (5 ms). The read time and hold time are both selected as 5 ms based on the real-time characteristics of the system. Specifically, this duration depends on the processing cycle of the ESP32 microcontroller and the conversion time of the ADS1115 ADC. The ADS1115 is configured at its maximum sampling rate of 860 samples per second (SPS). At the same time, this time interval ensures that each update of the PI controller parameters has sufficient time to take effect on the controlled system and accurately reflect changes in the output signal.

Table 1. Reward–state definition

Condition	Reward	State
r%<5% and error<1[V]	+20	S_1
r%>5% and error<1[V]	+10	S_2
r%<5% and error>1[V]	-5	S_3
r%>5% and error>1[V]	-15	S_4

4. EXPERIMENTAL SETUP

The experiments compare the system responses under different control configurations: open-loop control (Experiment **#1**), single-loop PI voltage feedback control (Experiment **#2**), dual-loop cascaded PI control with voltage and inductor current feedback (Experiment **#3**), adaptive RL-PI control trained for 200 Q-learning episodes (Experiment **#4**), and adaptive RL-PI control trained for 500 Q-learning episodes (Experiment **#5**) The training parameters for the RL-PI algorithm, including the learning rate α, discount factor γ, and exploration rate ε, are set to 0.64, 0.67, and 0.45, respectively. These values are consistently applied to both EX#4 and EX#5 to ensure a fair comparison. A summary of the experimental cases is provided in Table 2.

Table 2. Experiment illustrated

EXPERIMENT								
EX	NAME	Duty	K_{p1}	K_{i1}	K_{p2}	K_{i2}	Set point [V]	Episode
#1	Open loops	0.44					30	
#2	Single loops		2	50			30	
#3	Double loops		1.4	40	1.2	0	30	
#4	RL-PI						30	200
#5	RL-PI						30	500

An overview of the RL-PI model training system is illustrated in Figure 5. To reduce implementation cost, a 16-bit ADS1115 ADC is employed and interfaced with an ESP32 microcontroller via the $I2C$ protocol to acquire analog voltage and current measurements for the environment. The output voltage is measured using a resistive voltage divider. An ACS712-20A current sensor is utilized to measure the output current.

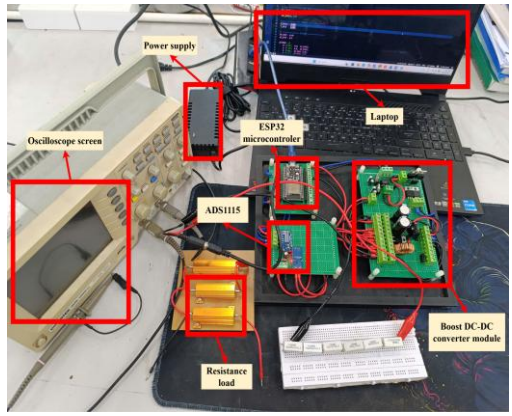

Fig. 5. Overview of the experimental setup and test scenarios

5. RESULTS

5.1. System response

We set the output load under open-circuit conditions to evaluate the system's overvoltage tolerance as the load condition approaches an open circuit. The system responses obtained from the oscilloscope when applying each controller in experiments EX#2-EX#5 are illustrated in Figure 6. In terms of steady-state error, the controller in EX#2 exhibits the largest error of approximately +2.8 V, with a 2% settling time of about ~0.45s. The result in EX#3 shows an improvement, where the steady-state error is reduced to +2.4 V (a decrease of 0.4 V compared EX#2), and the 2% settling time is shortened to approximately ~0.24s. This indicates that increasing the number of ideal integral and proportional components in EX#3 helps improve the system accuracy. For EX#4 and EX#5, the steady-state error is further reduced to approximately 0.8 V, representing a significant improvement of about 2 V compared to EX#2 and EX#3. The settling times for EX#4 and EX#5 are approximately 0.4s and 0.456s, respectively, showing no significant change compared to earlier experiments. This is because the proportional and ideal integral components in the PI controller do not substantially improve the system's settling time.

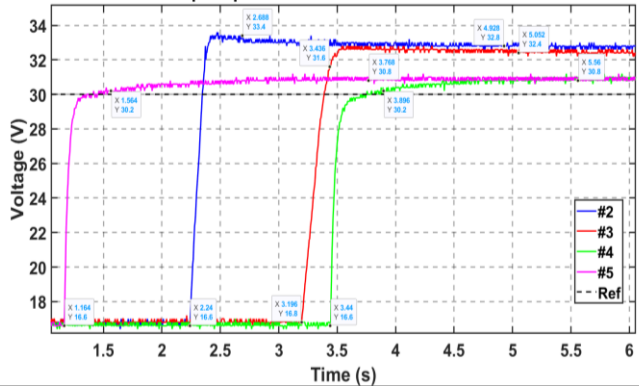

Fig. 6. System responses under open-circuit load conditions.
The output voltage tracks the reference of 30 V.

We conducted experiments to evaluate the steady-state error and output voltage ripple under various load resistance conditions. A summary of the measured steady-state output voltages for each experiment is presented in Table 3, where "OC" denotes the open-circuit condition at the output. It can be clearly observed that in EX#1, when the system is driven

by a PWM signal without feedback control, the output voltage error easily exceeds 5% in most cases with medium and high load resistance values. This indicates that the boost DC-DC converter cannot inherently maintain stability without a controller, due to the presence of a right-half-plane zero. The results from EX#4 and EX#5 show significant improvements, with the error maintained at around 3% under medium and high load conditions. Notably, under the open-circuit condition, EX#5 achieves a reduction in steady-state error of up to 7% compared to EX#2 and 5% compared to EX#3. Under the medium load condition (R = 470 Ω), EX#4 and EX#5 reduce the steady-state error by approximately 1.3% and 1% compared to EX#2 and EX#3, respectively.

Table 3. Controller performant comparison

	Vo [V]			Ripple [%]			Error [%]		
EX	R=940	R=470	OC	R=940	R=470	OC	R=940	R=470	OC
#1	41.4	32.8	95	0.97	2.44	0.84	38	9.3	216.67
#2	31.3	31.3	32.7	1.92	1.92	0.61	4.3	4.3	9
#3	31.3	31.2	32.4	1.28	1.28	1.23	4.3	4	8
#4	30.9	30.9	30.9	1.3	1.3	0.65	3	3	3
#5	31.8	30.9	30.9	1.3	1.3	1.3	2.7	3	3

5.2. Load adaptability

To evaluate the adaptability of the controller under resistive load variations that affect the output current of the boost converter, step changes in the output load resistance were introduced. Specifically, abrupt load variation experiments were conducted for the controllers in EX#2–EX#5. As observed in Figure. 7A, which illustrates the system responses of EX#2–EX#5 when the load resistance is suddenly changed from 290 Ω to 940 Ω, a clear difference in overshoot peaks can be seen. In EX#2 and EX#3, the peak voltages reach 35.2 V and 35.8 V, respectively. Meanwhile, the controllers in EX#4 and EX#5 show improved performance, with reduced overshoot peaks of 33.9V and 34V, respectively. When the load resistance is abruptly changed from 940 Ω to 290 Ω, as shown in Figure 7B, EX#2 and EX#3 exhibit voltage drops of −3.4V, whereas EX#4 and EX#5 achieve smaller undershoot magnitudes of −2.2 V and −2.4V, respectively. These results demonstrate that the RL-PI training process effectively enhances the system's ability to control both overshoot and undershoot. Table 4 summarizes the steady-state error according to the 2% criterion for the cases shown in Fig. 7. It can be observed that the system maintains the error below 5% of the reference value under all sudden load variation conditions. Compared to the performance summarized in Table 3, the controller configuration in EX#5 shows a slight increase in overshoot during load release. This behavior is consistent with the inherent characteristics of tuning the integral gain, as the policy derived from the Q-learning training process may favor higher K_i values, which can lead to increased overshoot in the system.

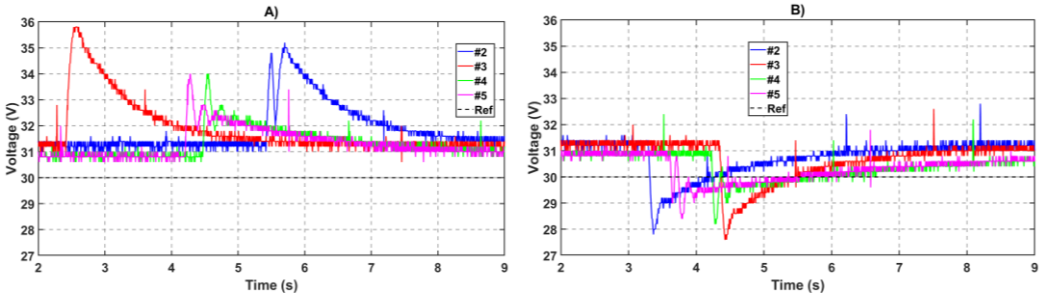

Fig. 7. System responses under load variations: (A) load change from 290 Ω to 940 Ω; (B) load change from 940 Ω to 290 Ω.

Table 4. System responses under load variations

	Vo [V]		POT [%]	
EX	940 to 290	290 to 940	940 to 290	290 to 940
#2	31.2	31.5	11	11.8
#3	31.05	31.2	11	14.7
#4	30.6	31	7.8	9.3
#5	30.6	31	7.2	9.7

6. CONCLUSION

In this study, an adaptive RL-PI controller was designed for a boost DC–DC converter operating under varying load conditions, with the training process conducted directly on real voltage disturbances. The proposed experimental framework employs computational hardware that is inexpensive and easy to deploy, making it suitable for future studies involving similar reinforcement learning–based training approaches. The experimental results confirm that the reinforcement learning–based adaptive controller is capable of optimizing the parameters of a conventional PI controller in real time. Furthermore, the results demonstrate the promising adaptability of reinforcement learning algorithms under practical operating conditions, achieving steady-state error below 5% and output voltage ripple below 2%.

Acknowledgement: This work was supported by Ho Chi Minh City University of Technology and Engineering, Vietnam for computational resources, and by the Artificial Intelligence for Automation and Control Laboratory (AICA Lab), Faculty of Electrical and Electronic Engineering, Ho Chi Minh City University of Technology and Engineering, Vietnam, for research facilities.

REFERENCES

[1] S. M. Ghamari, M. Hajihosseini, D. Habibi, and A. Aziz, “Design of an Adaptive Robust PI Controller for DC/DC Boost Converter Using Reinforcement-Learning Technique and Snake Optimization Algorithm,” *IEEE Access*, vol. 12, pp. 141814–141829, 2024, doi: https://doi.org/10.1109/ACCESS.2024.3440580.

[2] G. Nethaji and J. Kathirvelan, "Performance comparison between PID and fuzzy logic controllers for the hardware implementation of traditional high-voltage DC-DC boost converter," *Heliyon*, vol. 10, no. 17, Art. no. e36750, Sep. 2024, doi: https://doi.org/10.1016/j.heliyon.2024.e36750.

[3] S. W. Seo and H. H. Choi, “Digital implementation of fractional order PID-Type controller for boost DC-DC converter,” *IEEE Access*, vol. 7, pp. 142652–142662, 2019, doi: https://doi.org/10.1109/ACCESS.2019.2945065.

[4] W. R. Liou *et al*., “A Programmable Controller IC for DC/DC Converter and Power Factor Correction Applications,” *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2105–2113, 2013, doi: https://doi.org/10.1109/TII.2012.2235845.

[5] O. Djamel, G. Dhaouadi, S. Youcef, and M. Mahmoud, “Hardware Implementation of Digital PID Controller for DC-DC Boost Converter,” in *Proc. 4th Int. Conf. Power Electron. Appl. (ICPEA)*, Elazig, Turkey, 2019, pp. 1–4, doi: https://doi.org/10.1109/ICPEA1.2019.8911129.

[6] D. C. Meena and A. Devanshu, ‘‘Genetic algorithm tuned PID controller for process control,’’ *in Proc. Int. Conf. Inventive Syst. Control (ICISC),* Coimbatore, India, Jan. 2017, pp. 1–6, doi: https://doi.org/10.1109/ICISC.2017.8068639.

[7] K. J. Rathi and M. S. Ali, “Neural Network Controller for Power Electronics Circuits,” *IAES Int. J. Artif. Intell.*, vol. 6, no. 2, pp. 49–55, Jun. 2017, doi: https://doi.org/10.11591/ijai.v6.i2.pp49-55.

[8] M. H. Demir and M. Demirok, “Designs of Particle-Swarm-Optimization-Based Intelligent PID Controllers and DC/DC Buck Converters for PEM Fuel-Cell-Powered Four-Wheeled Automated Guided Vehicle,” *Appl. Sci.*, vol. 13, no. 5, Mar. 2023, doi: https://doi.org/10.3390/app13052919.

[9] M. Algreer, M. Armstrong, and D. Giaouris, “Adaptive PD+I control of a switch-mode DC-DC power converter using a recursive FIR predictor,” *IEEE Trans. Ind. Appl.*, vol. 47, no. 5, pp. 2135–2144, Sep. 2011, doi: https://doi.org/10.1109/TIA.2011.2161856.

[10] L. Ren, D. Wang, and Y. Zhang, “Adaptive Neural Network Control of Four-Switch Buck–B oost Converters,” *Actuators*, vol. 13, no. 9, Sep. 2024, doi: https://doi.org/10.3390/act13090375.

[11] . M. Cheng *et al*., “A Back Propagation Neural Network with Double Learning Rate for PID Controller in Phase-Shifted Full-Bridge Soft-Switching Power Supply,” *J. Electr. Eng. Technol.*, vol. 15, no. 6, pp. 2811–2822, Nov. 2020, doi: https://doi.org/10.1007/s42835-020-00523-5.

[12] N. T. T. Vu, H. D. Nguyen, and A. T. Nguyen, “Reinforcement Learning-Based Adaptive Optimal Fuzzy MPPT Control for Variable Speed Wind Turbine,” *IEEE Access*, vol. 10, pp. 95771–95780, 2022, doi: https://doi.org/10.1109/ACCESS.2022.3205124.

[13] N. P. Lawrence, M. G. Forbes, P. D. Loewen, D. G. McClement, J. U. Backstrom, and R. B. Gopaluni, “Deep Reinforcement Learning with Shallow Controllers: An Experimental Application to PID Tuning,” *Control Eng. Pract.*, vol. 118, Jan. 2022, doi: https://doi.org/10.1016/j.conengprac.2021.105046.

[14] Z. Guan and T. Yamamoto, “Design of a Reinforcement Learning PID Controller,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Glasgow, U.K., 2020, pp. 1–6, doi: https://doi.org/10.1109/IJCNN48605.2020.9207641.

[15] H. Song, Y. Liu, J. Zhao, J. Liu, and G. Wu, “Prioritized Replay Dueling DDQN Based Grid-Edge Control of Community Energy Storage System,” *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 4950–4961, Nov. 2021, doi: https://doi.org/10.1109/TSG.2021.3099133.

[16] S. Shahnooshi, J. Ebrahimi, and A. Bakhshai, “A Reinforcement Learning Controller Based on Double DQN for DC Microgrids with Constant Power Loads,” in *Proc. Can. Conf. Electr. Comput. Eng. (CCECE)*, 2024, pp. 376–380, doi: https://doi.org/10.1109/CCECE59415.2024.10667207.

[17] P. Kofinas and A. I. Dounis, “Fuzzy Q-learning agent for online tuning of PID controller for DC motor speed control,” *Algorithms*, vol. 11, no. 10, Oct. 2018, doi: https://doi.org/10.3390/a11100148.