IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 13 July 2024, accepted 18 August 2024, date of publication 26 August 2024, date of current version 3 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3448535

RESEARCH ARTICLE

Reinforcement Learning-Based Control of DC–DC Buck Converter Considering Controller Time Delay

DONGHUN LEE[1], BONGSEOK KIM[2], SOONHYUNG KWON[1], NGOC-DUC NGUYEN[1], MIN KYU SIM[2], (Member, IEEE), AND YOUNG IL LEE[1], (Senior Member, IEEE)

[1]Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul 01811, Republic of Korea
[2]Department of Data Science, Seoul National University of Science and Technology, Seoul 01811, Republic of Korea

Corresponding authors: Young Il Lee (yilee@seoultech.ac.kr) and Min Kyu Sim (mksim@seoultech.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2019R1A6A1A03032119 and Grant NRF-2021R1A6A1A03039981.

ABSTRACT Non-linearities and unmodeled dynamics in the control system inevitably degrade the quality and reliability of voltage stabilization performance in DC-DC buck converters. Reinforcement Learning (RL) is an emerging method to mitigate this issue. However, traditional RL typically necessitates significant computational resources and specialized processing units, thus being an economically unreasonable option. This paper proposes a high-performance RL-based method even suitable for a cost-effective Digital Signal Processor (DSP). To address the significant challenge of time delay in a DSP when training the RL agent, this paper adopts a Real-Time Deep Reinforcement Learning (RTDRL) approach that creates an augmented virtual decision process to eliminate the delay effect. The performance is validated through software simulation (PLECS) and an actual system, through which the proposed approach demonstrated superior performance compared to existing benchmarks, including existing approaches and artificial intelligence.

INDEX TERMS DC-DC synchronous buck converter, real-time deep reinforcement learning (RTDRL), digital signal processor (DSP), optimal control.

I. INTRODUCTION

With the widespread adoption of electric vehicles and advancements in renewable energy and storage technologies [1], there is a growing need for equipment related to diverse DC sources. In this context, DC-DC converters, utilized for converting direct current from one voltage level to another, are gaining attention in the industrial sector due to their simple structure and high efficiency. Accordingly, considerable research efforts are underway to attain superior performance in terms of stability [2] and efficiency.

Regarding control methods for DC-DC converters, proportional-integral-derivative (PID) [3] has been widely used for its easy implementation and wide applicability. Beyond PID, considerable research efforts have been made for better control methods. Mathematical model-based methods including model predictive control (MPC) [4] and sliding mode control (SMC) [5] allow for a precise understanding and interpretation of system behavior, thereby providing more reliable control. Nevertheless, the existence of non-linearities and unmodeled dynamics poses persistent and substantial challenges. These complexities complicate the design of control systems and unavoidably compromise stability and reliability [6].

To address this challenge, there have been notable interest and attention directed towards data-driven approach, as evidenced by recent developments [7]. Reinforcement Learning (RL) is emerging as a promising alternative in the field of data-driven control methods. The RL agent learns optimal policies through experience [8]. Specifically, Deep Reinforcement Learning (DRL), which adds deep neural networks to RL algorithms, has demonstrated significant potential in handling complex control tasks.

The associate editor coordinating the review of this manuscript and approving it for publication was Yuh-Shyan Hwang.

TABLE 1. Review of deep reinforcement learning-based converter control.

~~Other Paper~~	Actual system implementation	Current constraint	Controller (approx.cost)	RL algorithm	Standalone RL
C. Cui(2021) [9]	-	-	-	DQN	Yes
M. Andalibi(2021) [10]	-	-	-	DQN	No
J. He,(2021) [11]	-	-	-	DDPG	No
M. Gheisarnejad(2020) [12]	✓	-	MicroLabBox(10,000$)	DDPG	No
C. Cui(2022) [13]	✓	-	MicroLabBox(10,000$)	DQN	No
X. Meng(2022) [14]	✓	-	MicroLabBox(10,000$)	TD3	No
O. Zandi(2023) [15]	✓	-	PC(1,000$)	DQN	Yes
S. Saadatmand(2021) [16]	✓	✓	PC(1,000$)	DDPG	No
S. Kwon(2021) [17]	✓	✓	Ti28379D(100$)	DDPG	No
Proposed method(RTDRL)	✓	✓	Ti28379D(100$)	RTDRL	Yes

TABLE 2. Advantage and disadvantage between PI, MPC, DRL, and proposed RTDRL.

Type	Advantage	Disadvantage
PI	• executable with low-spec processor • easy to make and implement • can deal with delay problem	• difficulty to tune optimally
MPC	• optimal with perfect model • can deal with delay problem	• requires high-spec processor • high model dependency
DRL	• low model dependency • can learn optimal policies from complex environments	• can not deal with delay problem • requires high-spec processor • complexity in tuning hyper-parameter
RTDRL (Proposed)	• low model dependency • can learn optimal policies from complex environments • executable with low-spec processor • can deal with delay problem	• complexity in tuning hyper-parameter

Regarding the control issue for the non-linear and random behavior of the system, DRL can offer advantages in stability and fast response times without requiring prior knowledge of the model [10], [12], [13], [14], [15], [16], [17]. For example, the study [15] focuses on voltage control using multiple agents and demonstrates fast and stabilized results. Another study [13] addresses discrepancies between simulation models and real-world systems, making DRL more practical for applications. Often, DRL is combined with conventional approaches; for example, studies [16], [17] have introduced an additional proportional integral (PI) controller to regulate current constraints, or used DRL as a gain tuning tool for PI. Additionally, DRL is employed to seek adaptive horizons in the context of generalized predictive control [18].

While studies have yielded some promising outcomes, the DRL method confronts notable challenges that require attention before it can be regarded as a pragmatic alternative within the industry. We identify the following two main challenges: 1) DRL must be operable on low-cost and low-specification processing units, and 2) DRL must demonstrate high performance in actual device implementations.

Firstly, the challenge of incorporating DRL into industrial applications is primarily attributed to the requirement for high-spec processing units because recent developments in DRL heavily rely on deep neural networks with a large number of parameters [19], [20], demanding high-spec processing units for real-time operations. For instance, many studies have utilized tools such as MicroLabBox [12], [13], [14], [18], which can cost more than $10,000, or personal computers (PCs) [15], [16]. While these methods may offer high performance, their economic impracticality for industrial applications stems from the associated high costs and complexity.

Secondly, most existing studies are not validated through actual device implementation, being limited to simulation studies [10], [12]. The main obstacle preventing the widespread adoption of DRL in the industrial sector is its dependence on simulation-based learning methods [21]. Specifically, subtle differences between simulation environment and the actual operating environment, such as processing time or external disturbances, can significantly degrade the performance of trained agents [13], [22]. Therefore, consideration of these factors is essential for actual implementation. A comprehensive review of DRL-based converter control is presented in Table 1.

Consequently, this paper aims to develop a practical DRL agent with two primary goals: 1) to operate on low-spec processing units with limited computational resources, and 2) to demonstrate high performance in actual implementations beyond simulation.

Firstly, we limit the selection of controllers to low-spec processing units, specifically Digital Signal Processors (DSP), which are commonly accessible at a reasonable cost. Secondly, under the choice of a DSP controller, the most critical aspect of achieving high performance is the slow response time caused by the feedback delay in PWM (Pulse Width Modulation). Although updating the control after the actuator stabilizes can be one viable solution, this approach may hinder precise and fast control of the system. For example, the DSP's slow response time leads to performance limitations in existing DRL approaches [17], [23].

The time delay in feedback causes accumulation of errors, which hinders the effective achievement of voltage stabilization and compliance with current constraints, thereby deteriorating system reliability. While this is a challenge for all control systems, it is particularly critical for DRL

approaches. The root cause of this time delay issue is the inherent need of DRL for an immediate cause-and-effect relationship [8], [23]. That is, in situations involving time delays, relying solely on state information, which describes the context of the DRL controller, is insufficient for making optimal decisions.

To mitigate time delays, this paper utilizes a more proactive control method: the *Real-Time Deep Reinforcement Learning (RTDRL)* approach that creates an augmented virtual decision process to eliminate the delay effect. Specifically, the proposed method synchronizes the time step between actions and states by introducing a new augmented state that comprises a pair of the current state and preceding action, thereby providing sufficient information for decision-making and inferring the delay effect. This approach can be particularly useful in control systems constrained by real-time requirements and limited computational resources, offering a potentially cost-effective alternative. We summarize the advantages of this method over the popular existing methods in Table 2.

To achieve practical implementation in a Constant Power Load (CPL), this study follows a comprehensive sequential validation process, including software simulation based on PLECS, and the eventual realization of the system as a physical device. The empirical results demonstrate quicker transient times and less error compared to existing methods. The main contributions of this study can be summarized as follows.

- This paper presents an RTDRL approach that solves the inherent delay problem in DSP. The fundamental concept is to create an augmented virtual decision process to eliminate the delay effect.
- Our approach can deliver excellent performance at an affordable cost using DSP, requiring no additional techniques.
- Based on our proposed method, we design and implement a more practical DC-DC converter system. Our results demonstrate superior performance in both transient-time and steady-state control compared to existing methods.

II. PROBLEM FORMULATION AND PRELIMINARIES

A. PROBLEM FORMULATION

The circuit diagram of the DC-DC synchronous buck converter is illustrated in Fig. 1.

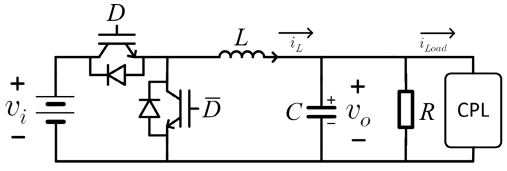


FIGURE 1. Synchronous buck converter circuit.

The parameters L, C, and R represent the inductance, capacitance, and resistance, respectively. A single control input, the duty ratio, manages both switches' pulse D and $\bar{D}$ in a complementary manner. v_o and v_i denote the output voltage and input voltage. i_L is the inductor current and i_{Load} is the current flowing through CPL. With the determined nominal voltage, the current flowing through CPL can be expressed as $i_{Load}(k) = P_{CPL}/v_o(k)$ for time k.

Contrary to the non-synchronous buck converter that operates in two modes, the synchronous buck converter always operates in Continuous Conduction Mode (CCM) [24], which facilitates the use of average model, as described by the following equations (1).

$$\frac{dx(k)}{dk} = A_c x(k) + B_c d(k) + E_c i_{Load}(k), \tag{1}$$

where

$$A_c = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix}, B_c = \begin{bmatrix} \frac{v_i}{L} \\ 0 \end{bmatrix}, E_c = \begin{bmatrix} 0 \\ -\frac{1}{C} \end{bmatrix}, x = \begin{bmatrix} i_L \\ v_o \end{bmatrix},$$

the expression for time k is applied where appropriate, and the duty ratio $d(k)$ is ranged between 0 and 1. The control objective of the DC-DC synchronous buck converter is to make the output voltage $v_o(k)$ as close as possible to the reference voltage v_{ref}, which is lower than the input voltage v_i, while maintaining the constraints on the inductor current during the transient response period. To convert this continuous-time model into a discrete-time state-space representation for controller design, the following discrete model is applied.

$$x(t+1) = Ax(t) + Bd(t) + Ei_{Load}(t), \tag{2}$$

where

$$A = e^{A_c T_s}, B = \int_0^{T_s} e^{A_c t} B_c dk, E = \int_0^{T_s} e^{A_c t} E_c dk,$$

Here, T_s is the sampling time, and t denotes the discrete time step.

B. FORMULATION AS AN MDP

This subsection formulates a Markov decision process (MDP) for controlling the DC-DC synchronous buck converter. An MDP is a mathematical framework designed for solving problems involving sequential decision-making under uncertainty. It is typically defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \gamma, \mathcal{P})$, where $\mathcal{S}$ represents the state space, $\mathcal{A}$ represents the action space, $\mathcal{R}$ represents the reward function $r(s, a)$: $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$ that determines the immediate reward value, γ is the discounted factor that balances the importance of immediate and future rewards, and $\mathcal{P}$ represents the probabilistic transition between the current state and the next state. Based on the system description by equation (2), the following components are defined.

1) STATE

The state at time t, $s_t \in \mathcal{S}$, is defined as follows:

$$s_t = [v_o(t-2), i_L(t-2), v_o(t-1), i_L(t-1), v_o(t), i_L(t), v_{ref}]. \tag{3}$$

Note that the state definition includes a history of observations capturing the temporal dynamics of the system, i.e., trends of voltage and current, providing the agent with context for better decision-making.

2) ACTION

The action at time t, $a_t \in \mathcal{A}$, is defined as the duty value $a_t = d(t) \in [0, 1]$. The duty value obtained from the action is used to create pulses through the PWM module to control the synchronous buck converter.

3) REWARD

The reward $r_t \in \mathcal{R}$ is defined as sum of reward for voltage alignment r_t^{vol} and reward for current limit r_t^{cur}:

$$r_t = r_t^{vol} + r_t^{cur}, \tag{4}$$

where

$$r_t^{vol} = \begin{cases} \beta_1 \cdot \dfrac{1}{|v_o(t) - v_{ref}|} & \text{if } |v_o(t) - v_{ref}| \le \eta \\ -\beta_2 \cdot |v_o(t) - v_{ref}| & otherwise \end{cases}, \tag{5}$$

$$r_t^{cur} = \begin{cases} -\beta_3 \cdot |i_L(t)| & \text{if } |i_L(t)| \ge \sigma \\ \beta_4 & otherwise \end{cases}, \tag{6}$$

where $\beta_i(>0)$, $i = 1, 2, \ldots, 4$ and $\eta(>0)$ are tuning parameters. Note that r_t^{vol} is a reward provided based on the output voltage's alignment with the difference from v_{ref}. The other term r_t^{cur} is defined to ensure that the inductor current $i_L(t)$ remains within the absolute limit of σ.

C. REINFORCEMENT LEARNING

RL is an algorithm for solving sequential decision-making problems that can be modeled as an MDP. The agent selects an action a_t based on a policy π, which maps from a state s_t to an action a_t, while the environment provides feedback in the form of a reward r_t and transitions to the next state s_{t+1}. The objective of RL is to determine the policy that maximizes the expected return, which is the discounted sum of future rewards, denoted by $G_t = \sum_{i=t}^{\infty} \gamma^{i-t} r_i$.

In RL, the action-value function $Q^\pi(s_t, a_t)$ is a fundamental component that captures the expected return associated with performing a specific action a_t for a given state s_t and then following a policy π thereafter. It can also be recursively defined using the Bellman equation as follows.

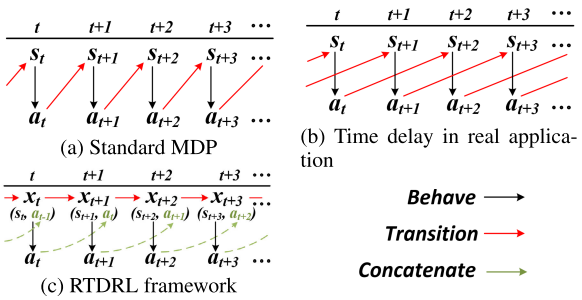
$$Q^\pi(s_t, a_t) = \mathbb{E}[G_t | s_t, a_t] = r_t + \mathbb{E}[Q^\pi(s_{t+1}, a_{t+1})] \tag{7}$$

This action-value function can be used to derive an optimal policy in a value-based approach by choosing the action with the highest expected return in each state [25], or it can be used implicitly in a policy-based approach [26]. The actor-critic method is a hybrid approach that combines both value-based and policy-based methods [20], [27]. It consists of two components: the actor and the critic, both of which are typically parameterized. The critic learns the action-value function $Q^\pi(s_t, a_t)$, while the actor updates the policy π_θ based on the critic's evaluation.

III. METHOD AND IMPLEMENTATION

A. TIME DELAY ISSUE IN MDP

In the framework of an MDP, it is typically posited that an action a_t directly impacts the ensuing next state s_{t+1}, signifying that a_t plays a critical role in shaping the subsequent next state s_{t+1}.



(a) Standard MDP (b) Time delay in real application (c) RTDRL framework

FIGURE 2. Time delay issue in MDP.

However, this presumption of immediate causality between action and the immediate next state may not be applicable in many engineering contexts. In many systems, the effect of an action a_t is observed with a temporal delay; that is, rather than affecting the immediate next state s_{t+1}, the action a_t affects on the later state s_{t+2}. This delayed interaction presents a nuanced challenge in accurately modeling such systems.

Fig. 2a and 2b contrast the two different scenarios. In the standard scenario of immediate impact, as depicted in Fig. 2a, the next state s_{t+1} is a function of the current state s_t and the current action a_t. In the scenario of delayed impact depicted in Fig. 2b, however, the next state s_{t+1} is a function of the current state s_t and the previous action a_{t-1}. This time delay is known to increase the difficulty of learning an optimal policy, as noted in [28].

The issue of a 1-step time delay also arises within the PWM peripheral of a DSP. This delay is attributed to the PWM mechanism, which ensures pulse stability by applying the duty value calculated from the current state at the start of the next carrier cycle, thereby generating the pulse.

B. REAL-TIME DEEP REINFORCEMENT LEARNING (RTDRL)

This study tackles the challenge of time delay in DSP by utilizing an RTDRL framework [23] based on the Soft Actor-Critic (SAC) algorithm [20]. In this framework, states and actions evolve simultaneously, enabling the agent to learn and act in real time. The underlying idea is to synchronize the time step between actions and states by introducing a new augmented state that includes a pair of current state and preceding action $x_t = (s_t, a_{t-1})$ as illustrated in Fig. 2c. Within this framework, agent policy π takes augmented state x_t instead of state s_t alone. The state alone does not provide sufficient information to accurately predict the future evolution of the system having delays. This requires modifying the conventional action-value function, originally

denoted as $Q^{\pi}(s_t, a_t)$, into the modified version described in equation (8). The augmented state, as a pair of the current state and preceding action $x_t = (s_t, a_{t-1})$, replaces the state s_t in the conventional action-value function. Note that the reward is also a function of the augmented state.

$$Q^{\pi}(x_t, a_t) = r(x_t) + \mathbb{E}[Q^{\pi}(x_{t+1}, a_{t+1})] \tag{8}$$

We utilize the SAC algorithm to derive 1) a modified optimal policy that maps the augmented state $x_t = (s_t, a_{t-1})$ to an optimal action a_t and 2) a Q-function $Q^{\pi}(x_t, a_t)$ in the RTDRL setting. The SAC algorithm is a state-of-the-art RL algorithm designed for continuous control. Its objective is to acquire an optimal policy that maximizes both the cumulative reward and the entropy of the policy, as described below.

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} (r_t + \alpha H(\pi(\cdot|s_t)))\right], \tag{9}$$

where π^* is the optimal policy, α is a hyperparameter that balances the importance of the reward and the entropy term, and $H(\pi(\cdot|s_t))$ is the entropy of the policy. Higher entropy promotes more exploration, potentially improving learning performance and preventing the policy from prematurely converging to local-optimal solutions [29].

To obtain an improved policy with an entropy-regularized objective function, an information projection is computed using the Kullback-Leibler (KL) divergence. This projection minimizes the divergence between the current policy and target distribution.

$$\pi_{new} = \arg\min_{\pi' \in \Pi} D_{KL}\left(\pi'(\cdot|s_t) \middle|\middle| \frac{\exp(Q(s_t, \cdot)/\alpha)}{Z(s_t)}\right), \tag{10}$$

where $Z(s_t)$ is the normalizing constant that ensures the policy distribution sums to one. Maximizing the expected value of the Q-function minus the entropy of the policy, the soft Q-function is equivalent to minimizing the KL divergence and can be formulated as

$$\mathbb{E}\left[Q(s_t, a_t) - \alpha \log \pi(a_t|s_t)\right] \tag{11}$$

The update procedure for combining RTDRL and SAC is outlined as follows.

1) Critic update: The critic network consists of two Q-function approximators Q_{ϕ_1} and Q_{ϕ_2} and the corresponding target network $Q_{\phi_1'}$ and $Q_{\phi_2'}$ for estimating the soft Q-function, $Q(x_t, a_t)$. It is updated by minimizing the loss function $J_Q(\phi_i)$.

$$J_Q(\phi_i) = \mathbb{E}[(Q_{\phi_i}(x_t, a_t) - y_t)]^2, \tag{12}$$

where

$$y_t = r_t + \gamma\Big(\min_{i\in\{1,2\}} Q_{\phi_i'}(x_{t+1}, \pi_\theta(x_{t+1})) - \alpha log\pi_\theta(a_{t+1}|x_{t+1})\Big). \tag{13}$$

2) Actor update: The policy network, $\pi_\theta(x_t)$ is updated to minimize the loss function $J_\pi(\theta)$.

$$J_\pi(\theta) = \mathbb{E}\Big[\alpha log\pi_\theta(a_t|x_t) - \min_{i\in\{1,2\}} Q_{\phi_i}(x_t, \pi_\theta(x_t))\Big] \tag{14}$$

Algorithm 1 Real Time Soft Actor-Critic

```
Initialize critic networks Q_φ1 and Q_φ2 corresponding target network, and actor network π_θ(x)
Initialize replay buffer D
for Every time step do
    Observe state s_t and previous action a_{t-1}
    Concatenate them to create the augmented state x_t = (s_t, a_{t-1})
    Select an action a_t from policy π_θ(x_t)
    Execute action a_t
    Observe a new state s_{t+1} and reward r_t = r(s_t, a_{t-1})
    Concatenate them to create augmented next state x_{t+1} = (s_{t+1}, a_t)
    Store [x_t, a_t, r_t, x_{t+1}] in the replay buffer D
    if time to update then
        for j in range (how many updates) do
            Randomly sample a batch of transition from D
            Update critic by gradient descent:
            φ_i ← φ_i − λ_Q ∇_{φ_i} J_Q(φ_i) for i ∈ {1, 2}
            Update policy by gradient descent:
            θ ← θ − λ_π ∇_θ J_π(θ)
            Update target network:
            φ_i' ← ρφ_i + (1 − ρ)φ_i' for i ∈ {1, 2}
        end for
    end if
end for
```

The pseudo-code for the Real-time Soft Actor-Critic is presented in Algorithm 1.

C. IMPLEMENTATION

This section outlines the implementation environment and the implementation procedure. The RTDRL learning environment for a PLECS-based DC-DC synchronous buck converter circuit controller is illustrated in Fig. 3. The simulation environment is configured to reflect the actual operating conditions as closely as possible. To emulate the typical delay in a DSP, this system integrates a pulse-delay module. Specifically, it has a 1-step delay of $200\mu s$ between the action execution and the response. To simulate the sensory margin of error, Gaussian noise is added to the sensed voltage and current: $N(0V, (0.025V)^2)$ is added to the voltage and $N(0A, (0.025A)^2)$ is added to the current. Communication between the RTDRL agent and PLECS is facilitated by a dynamic-link library (DLL) that acts as an interface to transfer action values and to receive the converter's state from PLECS.

The procedure for transitioning from the simulation system to the actual system is described below.

1) Train: Train the agent in simulation based on Fig. 3.

2) Test in simulation: Convert the agent's actor network to C language for DSP interfacing, and evaluate its learning effectiveness in simulation using PLECS.

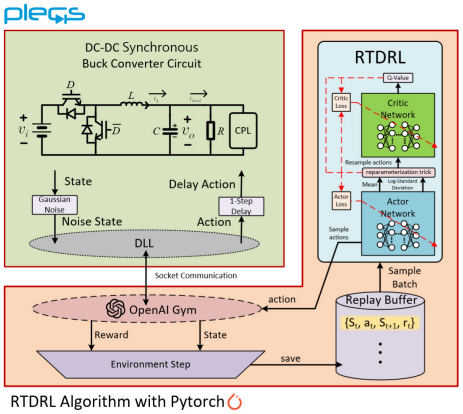


FIGURE 3. Setup for learning the RTDRL agent.

3) Test in actual system: After performing the simulation, the code is transferred to the DSP controller, followed by physical experiments conducted on a real DC-DC synchronous buck converter.

IV. SIMULATION AND EXPERIMENT RESULTS

A. EXPERIMENT SETUP

The experimental setup is depicted in Fig. 4. Power is transferred from the battery to the CPL using a DC-DC synchronous buck converter.

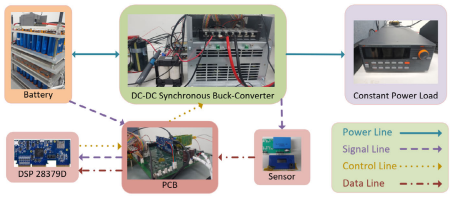


FIGURE 4. Actual-system experiment setting.

TABLE 3. DC-DC synchronous buck converter parameter.

Parameter	Value	Description
v_i	95-105V	Battery Input voltage
R	500Ω	Resistance
L	840μH	Inductance
C	4.7mF	Capacitance
f	10kHz	Switching frequency
T_s	200μs	Sampling time

This converter is regulated by a printed circuit board (PCB) board, which is connected to a TMS320F28379D controller. The PWM module, integrated into a DSP, generates a pulse for the converter at a frequency of 10 kHz, and our system's sampling time is 200μs. Specifically, the parameters for the DC-DC synchronous buck converter are listed in Table 3: the inductance L is 840μH, the capacitance C is 4.65mF, and the resistance R is 500Ω.

B. HYPER-PARAMETER SETUP

This subsection specifies the hyper-parameter setup for the baseline controller and the proposed RTDRL. Table 4

TABLE 4. Hyper-parameter of PI and MPC controller.

Algorithm	Parameter	value	Description
PI	(K_{pv}, K_{iv})	(4.26, 56.73)	PI gains of voltage control loop
	(K_{pi}, K_{ii})	(0.00782, 1.6145)	PI gains of current control loop
MPC	(K_1, L_1)	(-33.9e-3, -577.3e-6)	Current error gain
	(K_2, L_2)	(-8.0e-3, -333.9e-6)	Voltage error gain

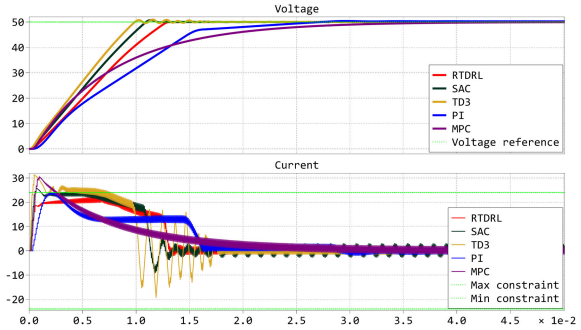
describes the hyper-parameters for two existing controllers, namely PI and MPC. Specifically, the system utilizes a dual-loop configuration with two PI controllers: an inner-loop current controller and an outer-loop voltage controller. These two PI controllers are initially tuned using MATLAB's automatic tuning tool to set baseline gain and then fine-tuned manually to find optimal gain. In the case of MPC, offline MPC was chosen instead of online MPC due to the computational burden on the DSP. Specifically, to enhance robustness to errors inherent in its offline nature, offline integral MPC was implemented. The gains of the offline integral MPC were obtained by solving a linear matrix inequality (LMI) based optimization problem [30].

The hyper-parameters for the proposed method are listed in Table 5, and those for other baseline RL-based controllers, such as normal Soft Actor-Critic (SAC) [20] and Twin Delayed Deep Deterministic policy (TD3) [19] are set identically. These hyper-parameters were determined through a grid search method, where the learning rate and reward tuning coefficients (η, σ, β_n) were systematically tested in approximately 200 combinations. To prevent potential DSP overload, the actor network employs a highly compact design, consisting of three hidden layers with ten neurons each. In contrast, the critic network, used only during training, comprises four hidden layers with 80 neurons each. The DRL training environment is set to a randomized scenario with the reference voltage ranging from 45V to 55V, input voltage ranging from 95V to 105V, and load power variation at either 0W or 500W. Each DRL controller was trained on 1 million scenarios.

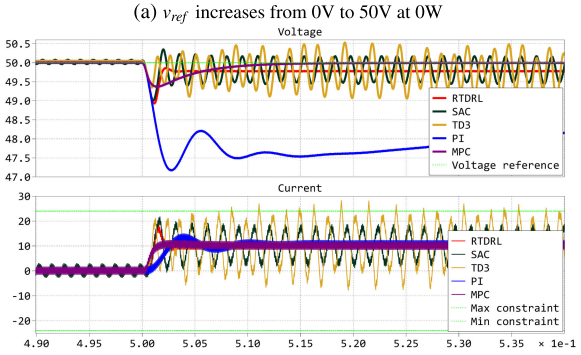
To simulate the sensory margin of error, Gaussian noises are added to sensed voltage, and current $N(0V, (0.025V)^2)$ is added to the voltage and $N(0A, (0.025A)^2)$ is added to the current.

TABLE 5. Hyper-parameter of RTDRL.

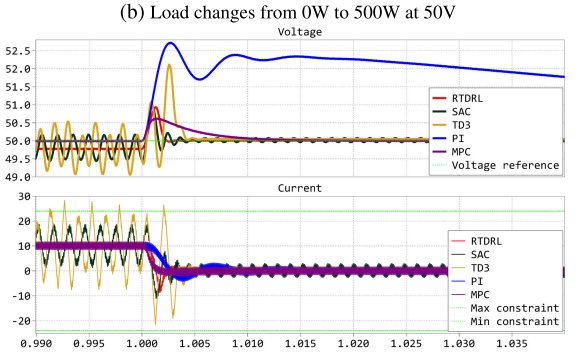
Parameter	Value	Description
λ	0.0003	Learning rate
γ	0.99	Discount factor
η	10	Sub-goal of r^{vol}
σ	24	Sub-goal of r^{cur}
β_1, β_2	2, -5	r^{vol} tuning coefficient
β_3, β_4	10, 51	r^{cur} tuning coefficient
AN	[10, 10, 10]	Actor network size
CN	[80, 80, 80, 80]	Critic network size



(a) v_{ref} increases from 0V to 50V at 0W



(b) Load changes from 0W to 500W at 50V



(c) Load changes from 500W to 0W at 50V

FIGURE 5. Comparison of simulation test results: each figure shows the results with v_{ref} at 50V.

C. SIMULATION AND EXPERIMENT RESULT

This subsection presents the results of both the simulation and actual-system tests, following the procedure discussed in Section III-C. The primary evaluation criteria involve assessing the performance of voltage tracking and ensuring compliance with current constraints under the following conditions.

- Voltage reference variation: (0V → 50V, 45V → 55V, and 55V → 45V) at 0W and 500W.
- Load power variation: (0W → 500W, 500W → 0W) with voltage reference levels of 45V, 50V, and 55V.

1) SIMULATION RESULT

Fig. 5 presents the simulation results under various voltage/load conditions. The proposed RTDRL demonstrates superior voltage tracking performance, achieving the reference voltage faster than existing methods such as PI and MPC, as shown in Fig. 5a. Furthermore, it must be emphasized that addressing time delay issues is essential for DRL-based methods. The standard DRL method shows some instability in its current control results. These issues become more pronounced in scenarios with load variation, as demonstrated in Fig. 5b-5c. Notably, standard DRL methods, lacking the capability to handle time delays, exhibit significant oscillation and instability compared to other controllers, suggesting their unsuitability for use in DSP settings.

2) EXPERIMENT RESULTS

Physical tests are conducted with actual systems to validate the effectiveness of the proposed method. The first experiment focused on the voltage-tracking performance, taking into account the current constraint and transient time. The second experiment assessed the algorithm's robustness under varying load conditions, with a particular focus on the transient errors. Fig. 6 shows the results of the voltage reference-tracking experiments, and the algorithms' performance is summarized in Table 6.

The rise/fall time is defined as the time required for the voltage to increase or decrease from 10% to 90% of the steady-state value, starting from the initial value. The results indicate that the proposed RTDRL agent exhibited the fastest speed and achieved stable convergence in all cases while ensuring the current constraints of ±24A, indicated by a blue dotted line. Specifically, the entire DRL controller demonstrated advantages by achieving faster rise/fall times compared to existing methods such as MPC or PI, as shown in Table 6.

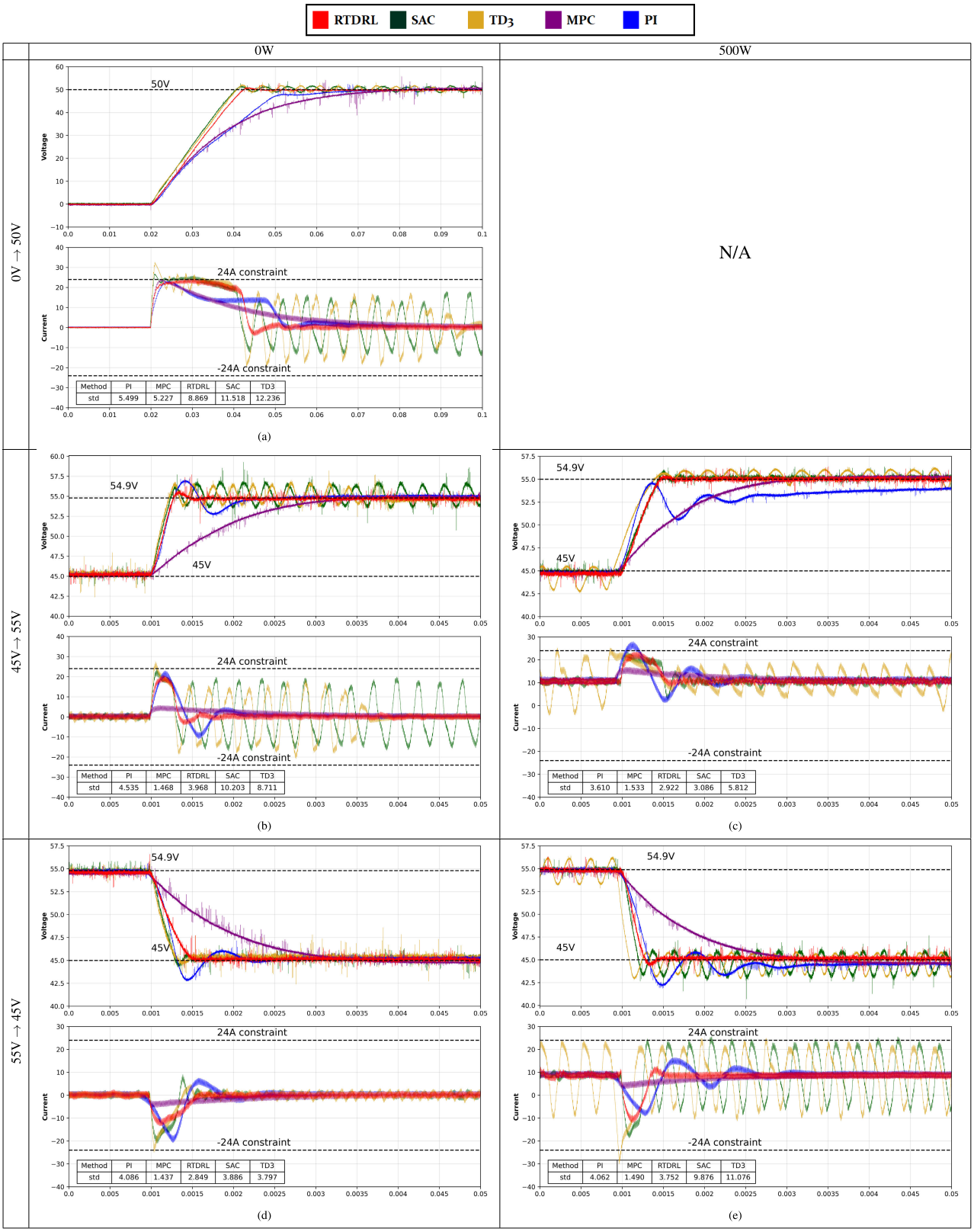
TABLE 6. Performance of meeting current constraints and rise/fall time.

Load power	Target voltage	RTDRL	SAC	TD3	MPC	PI
0W	0V → 50V	**Satisfied (8.43ms)**	Unsatisfied (8.05ms)	Unsatisfied (8.23ms)	Satisfied (14.98ms)	Satisfied (12.27ms)
	45V → 55V	Satisfied (1.702ms)	**Satisfied (1.61ms)**	Unsatisfied (1.60ms)	Satisfied (13.90ms)	Satisfied (1.917ms)
	55V → 45V	Satisfied (2.90ms)	**Satisfied (2.02ms)**	Satisfied (2.10ms)	Satisfied (11.20ms)	Satisfied (2.53ms)
500W	45V → 55V	**Satisfied (3.34ms)**	Satisfied (3.50ms)	Unsatisfied (4.00ms)	Satisfied (12.89ms)	Unsatisfied (2.20ms)
	55V → 45V	Satisfied (1.70ms)	**Satisfied (1.53ms)**	Unsatisfied (1.73ms)	Satisfied (14.40ms)	Satisfied (2.75ms)

*Values in parenthesis indicate rise/fall time

However, as observed in simulations, the performance degradation due to the time delay caused by the DSP suggests a limitation of standard DRL. For example, compared to the proposed RTDRL method, TD3 and SAC exhibited considerable oscillation in the current/voltage control results and failed to meet the current constraints, as illustrated in Figs. 6a, 6b, and 6e. This highlights the significance of addressing the feedback delay issue in the PWM of a DSP to achieve optimal performance in real-world applications.

Fig. 7 shows the results of the second experiment, which evaluates performance under load power variation, while Fig. 8 presents a comparison of the transient errors, measured as Integral Square Error (ISE), Integral Absolute Error (IAE), and Root Mean Square Error (RMSE). These values quantify the controller's deviation from the reference values for a given



Method	PI	MPC	RTDRL	SAC	TD3
std	5.499	5.227	8.869	11.518	12.236

Method	PI	MPC	RTDRL	SAC	TD3
std	4.535	1.468	3.968	10.203	8.711

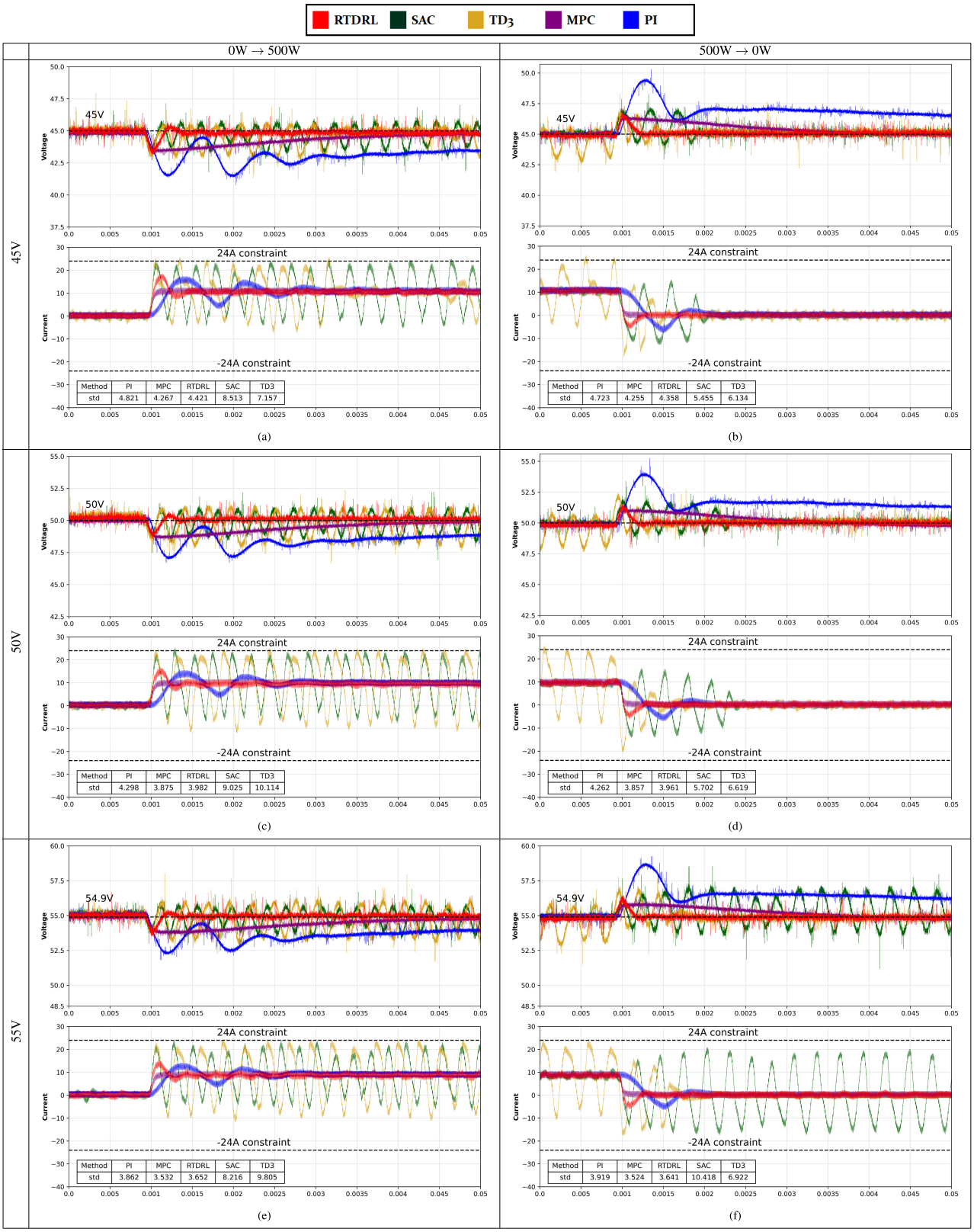
Method	PI	MPC	RTDRL	SAC	TD3
std	3.610	1.533	2.922	3.086	5.812

Method	PI	MPC	RTDRL	SAC	TD3
std	4.086	1.437	2.849	3.886	3.797

Method	PI	MPC	RTDRL	SAC	TD3
std	4.062	1.490	3.752	9.876	11.076

FIGURE 6. Comparison of experimental results with changing the reference voltage under CPL.

time interval. Based on these findings, the suggested RTDRL method presents a promising alternative to the conventional advanced approaches. Particularly in scenarios with load variation, traditional PI controllers demonstrate slow



Method	PI	MPC	RTDRL	SAC	TD3
std	4.821	4.267	4.421	8.513	7.157

Method	PI	MPC	RTDRL	SAC	TD3
std	4.723	4.255	4.358	5.455	6.134

Method	PI	MPC	RTDRL	SAC	TD3
std	4.298	3.875	3.982	9.025	10.114

Method	PI	MPC	RTDRL	SAC	TD3
std	4.262	3.857	3.961	5.702	6.619

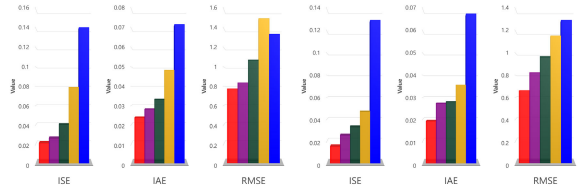
Method	PI	MPC	RTDRL	SAC	TD3
std	3.862	3.532	3.652	8.216	9.805

Method	PI	MPC	RTDRL	SAC	TD3
std	3.919	3.524	3.641	10.418	6.922

FIGURE 7. Comparison of experimental results under varying load power conditions with CPL.

voltage recovery and the largest transient errors, as shown in Fig. 8.

Additionally, classic DRL-based controllers, such as TD3 and SAC, are observed to have significant oscillation issues



	ISE	IAE	RMSE
RTDRL	0.021	0.024	0.773
MPC	0.026	0.028	0.837
SAC	0.040	0.033	1.074
TD3	0.079	0.048	1.505
PI	0.141	0.072	1.343

	ISE	IAE	RMSE
RTDRL	0.015	0.019	0.659
MPC	0.025	0.027	0.825
SAC	0.033	0.028	0.972
TD3	0.047	0.036	1.159
PI	0.130	0.068	1.301

(a) Load changes from 0W to 500W (b) Load changes from 500W to 0W

FIGURE 8. Error performance of the controller.

in current control during load variation settings due to the accumulation of errors caused by time delays, as illustrated in Figs. 7a, 7c, and 7e. In each figure, standard deviations (marked as "std") for the five control methods are presented in a small table, in which the larger standard deviations indicate more unstable performance. Our experimental results show that TD3 and SAC do not converge in most scenarios, exhibiting the largest standard deviations. The root cause of these unstable control outcomes in classic DRL-based controllers is that the reinforcement learning method inherently requires an immediate cause-and-effect relationship, known as the Markov property. This means the present state must have complete information for the DRL controller's decision-making. In other words, in situations involving time delays when using DSP, relying solely on controller's current state is not sufficient for making optimal decisions.

In contrast, the proposed RTDRL method resolves this issue by creating an augmented virtual decision process to mitigate the impact of delays, exhibiting a small standard deviation similar to existing controllers such as PI and MPC, and showing stable control. The proposed method demonstrated minimal error across all scenarios and exhibited superior performance, surpassing even advanced control methods such as MPC.

These results highlight the effectiveness of RTDRL in control systems operating under real-time requirements and limited computing resources. The proposed method can be implemented using ordinary DSP controllers, offering superior performance at a reasonable cost.

V. CONCLUSION

This study proposes a high-performance DRL-based controller for a DC-DC synchronous buck converter that utilizes a cost-effective DSP. To address the issue of time-delay, the proposed controller undergoes training within an RTDRL framework, establishing an augmented virtual decision process to eliminate the impact of delays effectively. Moreover, the controller simultaneously considers both voltage control and current constraints, guaranteeing the controller's reliability for actual systems. The results demonstrate improvements in transient time and steady-state error while maintaining current constraints, in comparison to existing methods. For future work, the proposed method can be extended to other types of DC-DC power converters (e.g., Dual Active Bridge converters and DC-DC buck-boost power converters), aiming to become more generalized methods that enhance the overall performance and efficiency of the control systems at a reasonable cost.

ACKNOWLEDGMENT

(Donghun Lee and Bongseok Kim are co-first authors.)

REFERENCES

[1] O. P. Jaga, R. Gupta, B. Jena, and S. GhatakChoudhuri, "Bi-directional DC/DC converters used in interfacing ESSs for RESs and EVs: A review," *IETE Tech. Rev.*, vol. 40, no. 3, pp. 334–370, May 2023.

[2] S. Sumsurooah, M. Odavic, S. Bozhko, and D. Boroyevich, "Robust stability analysis of a DC/DC buck converter under multiple parametric uncertainties," *IEEE Trans. Power Electron.*, vol. 33, no. 6, pp. 5426–5441, Jun. 2018.

[3] A. Visioli, *Practical PID Control* (Advances in Industrial Control). London, U.K.: Springer, 2006.

[4] P. Karamanakos, E. Liegmann, T. Geyer, and R. Kennel, "Model predictive control of power electronic systems: Methods, results, and challenges," *IEEE Open J. Ind. Appl.*, vol. 1, pp. 95–114, 2020.

[5] Z. Wang, S. Li, and Q. Li, "Discrete-time fast terminal sliding mode control design for DC–DC buck converters with mismatched disturbances," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1204–1213, Feb. 2020.

[6] J. Zhai, H. Wang, J. Tao, and Z. He, "Observer-based adaptive fuzzy finite time control for non-strict feedback nonlinear systems with unmodeled dynamics and input delay," *Nonlinear Dyn.*, vol. 111, no. 2, pp. 1417–1440, Jan. 2023.

[7] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2042–2062, Jun. 2018.

[8] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, vol. 135. Cambridge, MA, USA: MIT Press, 1998.

[9] C. Cui, N. Yan, B. Huangfu, T. Yang, and C. Zhang, "Voltage regulation of DC–DC buck converters feeding CPLs via deep reinforcement learning," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 3, pp. 1777–1781, Mar. 2022.

[10] M. Andalibi, M. Hajihosseini, S. Teymoori, M. Kargar, and M. Gheisarnejad, "A time-varying deep reinforcement model predictive control for DC power converter systems," in *Proc. IEEE 12th Int. Symp. Power Electron. for Distrib. Gener. Syst. (PEDG)*, Jun. 2021, pp. 1–6.

[11] J. He, L. Xing, and C. Wen, "Weighting factors' real-time updating for finite control set model predictive control of power converters via reinforcement learning," in *Proc. IEEE 16th Conf. Ind. Electron. Appl. (ICIEA)*, Aug. 2021, pp. 707–712.

[12] M. Gheisarnejad, H. Farsizadeh, and M. H. Khooban, "A novel nonlinear deep reinforcement learning controller for DC–DC power buck converters," *IEEE Trans. Ind. Electron.*, vol. 68, no. 8, pp. 6849–6858, Aug. 2021.

[13] C. Cui, T. Yang, Y. Dai, C. Zhang, and Q. Xu, "Implementation of transferring reinforcement learning for DC–DC buck converter control via duty ratio mapping," *IEEE Trans. Ind. Electron.*, vol. 70, no. 6, pp. 6141–6150, Jun. 2023.

[14] X. Meng, Y. Jia, Q. Xu, C. Ren, X. Han, and P. Wang, "A novel intelligent nonlinear controller for dual active bridge converter with constant power loads," *IEEE Trans. Ind. Electron.*, vol. 70, no. 3, pp. 2887–2896, Mar. 2023.

[15] O. Zandi and J. Poshtan, "Voltage control of a quasi Z-source converter under constant power load condition using reinforcement learning," *Control Eng. Pract.*, vol. 135, Jun. 2023, Art. no. 105499.

[16] S. Saadatmand, P. Shamsi, and M. Ferdowsi, "Adaptive critic design-based reinforcement learning approach in controlling virtual inertia-based grid-connected inverters," *Int. J. Electr. Power Energy Syst.*, vol. 127, May 2021, Art. no. 106657.

[17] S. Kwon, C. Yoon, and Y.-I. Lee, "Practical implementation method of reinforcement learning for power converter," *IFAC-PapersOnLine*, vol. 55, no. 9, pp. 437–441, 2022.
[18] C. Cui, Y. Dong, X. Dong, C. Zhang, and A. M. Y. M. Ghias, "Adaptive horizon seeking for generalized predictive control via deep reinforcement learning with application to DC/DC converters," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 71, no. 5, pp. 2217–2228, May 2024.
[19] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1587–1596.
[20] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," 2018, *arXiv:1812.05905*.
[21] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[22] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: A survey," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2020, pp. 737–744.
[23] S. Ramstedt and C. Pal, "Real-time reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–10.
[24] R. Nowakowski and N. Tang, "Efficiency of synchronous versus non-synchronous buck converters," Texas Instrum. Incorporated, Tech. Rep. SLYT358, 2009.
[25] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[26] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2000, pp. 1057–1063.
[27] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.
[28] J. B. Travnik, K. W. Mathewson, R. S. Sutton, and P. M. Pilarski, "Reactive reinforcement learning in asynchronous environments," *Frontiers Robot. AI*, vol. 5, p. 79, Jun. 2018.
[29] J. Schulman, X. Chen, and P. Abbeel, "Equivalence between policy gradients and soft Q-learning," 2017, *arXiv:1704.06440*.
[30] S. Kim, J. Kim, C. R. Park, and Y. I. Lee, "Output-feedback model predictive controller for voltage regulation of a DC/DC converter," *IET Control Theory Appl.*, vol. 7, no. 16, pp. 1959–1968, Nov. 2013.

DONGHUN LEE received the B.S. degree in electrical and information engineering from Seoul National University of Science and Technology (SeoulTech), Seoul, South Korea, in 2023, where he is currently pursuing the M.S. degree in electrical and information engineering. His research interests include applications of reinforcement learning in power electronic systems.

BONGSEOK KIM received the B.S. degree in industrial engineering and the M.S. degree in data science from Seoul National University of Science and Technology (SeoulTech), Seoul, South Korea, in 2021 and 2024, respectively. His research interests include applications of reinforcement learning and artificial intelligence in energy systems.

SOONHYUNG KWON received the B.S. and M.S. degrees in electrical and information engineering from Seoul National University of Science and Technology (SeoulTech), Seoul, South Korea, in 2021 and 2024, respectively. His research interests include power electronics and smart grids.

NGOC-DUC NGUYEN received the B.Sc. degree in mechatronics from Ho Chi Minh University of Technology, Vietnam, in 2014, and the M.S. and Ph.D. degrees in control engineering from Korea Maritime and Ocean University, South Korea, in 2016 and 2019, respectively. He was a Postdoctoral Researcher with the Research Center for Electrical and Information Technology, Seoul National University of Science and Technology (SeoulTech). His research interests include automatic control implementation on robotics, electrical machines, power converters, hybrid energy storage systems, and energy management strategies for microgrids.

MIN KYU SIM (Member, IEEE) received the Ph.D. degree in industrial engineering from Georgia Institute of Technology, Atlanta, GA, USA, in 2014. From 2015 to 2017, he was a Portfolio Manager and a Quantitative Researcher with asset management firms. From 2018 to 2019, he was a Research Professor with the Smart Energy Research Center, Kyung Hee University, South Korea. Since September 2019, he has been an Associate Professor with the Department of Data Science and the Department of Industrial Engineering, Seoul National University of Science and Technology (SeoulTech), Seoul, South Korea. His research interests include stochastic process application, smart-grid operation, reinforcement learning-based optimal control, and quantitative finance.

YOUNG IL LEE (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in control and instrumentation engineering from Seoul National University (SNU), in 1986, 1988, and 1994, respectively. He was a Visiting Research Fellow with the Department of Engineering Science, Oxford University, from 1998 to 1999 and in 2007. He was with Gyeongsang National University, from 1994 to 2001, and moved to Seoul National University of Science and Technology (SeoulTech), in 2001. He is currently a Professor with the Department of Electrical and Information Engineering and the Director of the Research Center of Electrical and Information Technology, SeoulTech. His scientific research interests include MPC for systems with input constraints and model uncertainties, MPC method for various converters and inverters, control of EV chargers, control of AC motors for EV application, and energy management algorithm of microgrids. He has served as an Editor for *International Journal of Control, Automation and Systems* and *International Journal of Automotive Technology*, from 2017 and 2019, respectively.