# Implementation of Transferring Reinforcement Learning for DC–DC Buck Converter Control via Duty Ratio Mapping

Chenggang Cui, *Member, IEEE*, Tianxiao Yang, *Student Member, IEEE*, Yuxuan Dai, Chuanlin Zhang, *Senior Member, IEEE*, and Qianwen Xu, *Member, IEEE*

***Abstract*—The reinforcement learning (RL) control approach with application to power electronics systems has become an emerging topic, while the sim-to-real issue remains a challenging problem as very few results can be referred to in the literature. Indeed, due to the inevitable mismatch between simulation models and real-life systems, offline-trained RL control strategies may sustain unexpected hurdles in practical implementation during the transfer procedure. In this article, a transfer methodology via a delicately designed duty ratio mapping is proposed for a dc–dc buck converter. Then, a detailed sim-to-real process is presented to enable the implementation of a model-free deep reinforcement learning controller. As the main contribution of this article, the proposed methodology is able to endow the control system to achieve: 1) voltage regulation and 2) adaptability and optimization abilities in the presence of uncertain circuit parameters and various working conditions. The feasibility and efficacy of the proposed methodology are demonstrated by comparative experimental studies.**

***Index Terms*—DC–DC buck converter, deep reinforcement learning (DRL), duty ratio mapping (DRM), practical implementation.**

## NOMENCLATURE

| | |
|---|---|
| $\beta_1, \beta_2$ | Reward coefficients. |
| $\beta_3$ | Penalty coefficient. |
| $\epsilon_1, \epsilon_2$ | Subgoals of the expected error. |
| $\frac{de(t)}{dt}$ | Time derivative of the tracking error. |
| $\frac{dv_o(t)}{dt}$ | Time derivative of the output voltage. |
| $\gamma$ | Discount factor, $\gamma \in [0, 1]$ |
| $\varepsilon$ | Exploration rate. |
| $A$ | Actions, $a \in A$ |
| $a_{t-\text{ran}}$ | Random action. |
| $C$ | Output capacitance. |
| $d$ | Duty ratio of the dc–dc buck converter. |
| $D_{\text{real}}$ | Information table of experiments. |
| $D_{\text{sim}}$ | Information table of simulations. |
| $E$ | Input dc source. |
| $e$ | Tracking error of the output voltage. |
| $e_{\text{del}}$ | Delay signal of the tracking error. |
| $f$ | Mapping between the duty ratio in the simulation and real environment. |
| $i_{\text{CPL}}$ | Current of CPL. |
| $i_L$ | Inductance current. |
| $L$ | Input inductance. |
| $P$ | State transition probability. |
| $p$ | Random value between 0 and 1 about choosing action. |
| $P_{\text{CPL}}$ | Power of CPL. |
| $R$ | Reward function. |
| $S$ | States, $s \in S$ |
| $v_{o\_\text{del}}$ | Delay signal of the output voltage. |
| $v_o$ | Output voltage. |
| $V_{\text{ref}}$ | Nominal voltage. |

Manuscript received 25 January 2022; revised 4 May 2022 and 20 June 2022; accepted 12 July 2022. Date of publication 26 July 2022; date of current version 23 January 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 51607111 and Grant 62173221, and in part by Shanghai Rising-Star Program under Grant 20QA1404000. *(Corresponding author: Chuanlin Zhang.)*

Chenggang Cui, Tianxiao Yang, Yuxuan Dai, and Chuanlin Zhang are with the Intelligent Autonomous Systems Laboratory, College of Automation Engineering, Shanghai University of Electric Power, Shanghai 200090, China (e-mail: cgcui@shiep.edu.cn; tianxiaoy9@mail.shiep.edu.cn; yuxuandai@mail.shiep.edu.cn; clzhang@shiep.edu.cn).

Qianwen Xu is with the Electric Power and Energy Systems Division, KTH Royal Institute of Technology, 114 28 Stockholm, Sweden (e-mail: qianwenx@kth.se).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIE.2022.3192676.

Digital Object Identifier 10.1109/TIE.2022.3192676

## I. INTRODUCTION

THE DC microgrid is becoming more and more attractive due to its conspicuous features compared with the ac microgrid, such as simple structure, easy control, strong robustness, and environmental friendliness. In the future, the power-electronics-based dc power system will possibly become a dominating form in national grids. Regarding the control issue for the dc power electronics systems, it is well acknowledged that several common challenges exist, which include the following.

1) Many nonlinear phenomena such as bifurcations and chaotic behavior occur in dc–dc converters mainly due to the switching action among all the different topologies of the circuit [1].
2) In order to ensure the stability of the system and detect the real-time status, the protection circuit and the sampling circuit are designed, respectively, which are mostly been neglected in the controller design procedure [2].

TABLE I
SKETCH COMPARISON BETWEEN DRL, MPC, AND PI CONTROLLERS

| | PI controller | Direct MPC | Explicit MPC | DRL controller |
|---|---|---|---|---|
| **Performance** | Not optimal | Optimal with perfect model | Optimal with perfect model | Close to optimal |
| **Online computation cost** | Low | Very high | High | Low |
| **Offline computation cost** | Model identification | Model identification | Model identification | Policy and model identification |
| **Reliance on model** | Need for parameter tuning | At all times | At all times | Only for training |
| **Sensitivity to tuning** | High | High | High | Low |
| **Processor required** | DSPs/FPGAs/MCU... | Not suitable for application | DSPs/FPGAs/MCU... | DSPs/FPGAs/MCU... |

3) Switching frequencies of power converters have been significantly increased to enhance the power density, which implies that the influences of radiated electromagnetic interference (EMI) are more and more serious [3].

Therefore, the problem of regulating power electronics systems with high performance has been a subject of great interest in recent years and various model-driven control strategies have been proposed.

However, it should be noted that the aforementioned factors would lead to the fact that classical model-driven control methods, such as proportional-integral-derivative (PID) control [4], model predictive control (MPC) [5], sliding mode control [6], composite control [7], etc., may exhibit slow dynamic response speed, output waveform distortion, or large fluctuation of the circuit states [8]. These model-based methods may show the adaption ability to the working condition variations, however, they might be very difficult to adapt to the different circuit component parameters. This means that when the system parameters are abruptly changed or the components are replaced, more efforts need to be spared to redesign the control scheme or control parameters in order to adapt to the new platform.

Aiming to present a more effective stabilization result for power electronics systems, in recent years, intelligent control methods for dc–dc converters have become a trend and attracted considerable attention from both industrial and scientific communities; see, for instance, [9]–[11], to mention only a few. Sketchily, they can be classified into three categories: intelligent model-driven methods [12], data-driven methods [13], and hybrid methods [14]. As a typical data-driven control strategy, recent literature has shown that deep reinforcement learning (DRL) has been gradually applied to the advanced control issue for power electronics systems [15]. The main idea of DRL is that the agent searches for an optimal policy to make decisions by interacting with the external environment [16]. Xia et al. [17] design a distributed multiagent DRL controller for the islanded dc microgrid and demonstrate the effectiveness via simulation. A DRL controller based on a fuzzy system is proposed in [18] to increase the stability of the frequency control subsystem in a microgrid. An adaptive data-driven method based on the ADRC strategy is adopted in [19] to build a programmable grid with a single voltage bus by dc–dc converters. However, these methods are mainly deployed in simulation setup or partly model-based, while very few previous contributions have been dedicated to the practical implementation of pure data-driven DRL methodology into a real-life power electronics system.

In terms of application, the computational complexity of the algorithm is one of the key issues to be considered. Although some of the advanced control methods may provide desirable control performance for large-signal fluctuations, they tend to be difficult to implement due to the long processing time. For example, the online computation time, especially for MPC, may be infeasible for large systems and/or for systems with long control horizons [20]. Table I analyzes the processing time, the sort of processor required, the advantages and disadvantages of the classical PI control [21], a set of MPC method [22], [23], and DRL control [24]. These comparative results have also demonstrated the potential of implementing DRL algorithms into practice.

Therefore, the transfer of DRL algorithms from simulation to implementation is of practical significance from an industrial application point of view. However, as summarized in [25], the transfer methods may cause several new challenges. Specifically, the gap between the simulation and implementation degrades the performance of the trained policies as the models are utilized in the real-life system. Therefore, extensive efforts are conducted to reduce the sim-to-real gap and accomplish more efficient policy transfer. To some extent, in the real-life dc–dc circuit systems, the sampling modules, the protection circuits, and unmodeled dynamics or other external disturbances can be considered as the main influencing factors, which can lead to a large steady-state regulation error in practice.

Regarding the transfer issues of DRL from simulation to practice, one of the most widely used methods is transfer learning (TL), which utilizes external experience from tasks to alleviate the burden of learning. The application of TL involves various dimensions, such as reward shaping, intertask mapping, policy transfer, etc. [26]. Book et al. [27] introduce a transfer method from the offline simulation to the online training and inference on real motor drive systems. A new algorithm called SPOTA to learn a control policy exclusively from a randomized simulator without using any data from the simulator is proposed in [28]. However, regarding the application of the DRL algorithm into practical dc–dc converters, there are very few existing results

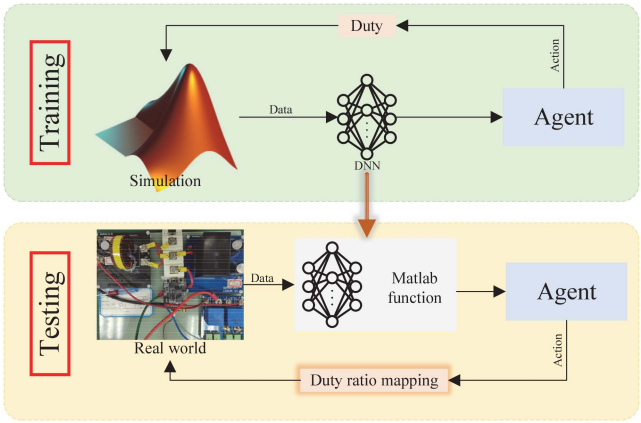


Fig. 1. Proposed transfer procedure.

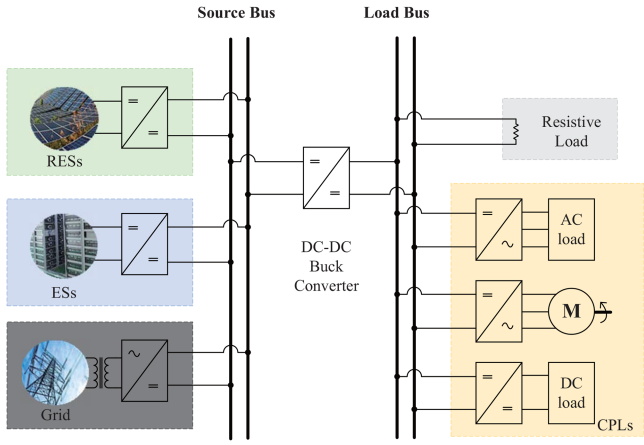


Fig. 2. General layout of a typical dc microgrid.

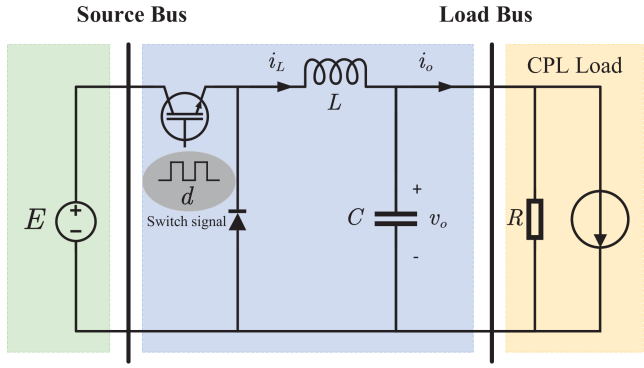


Fig. 3. Topology of a dc–dc buck converter.

that can be found in the literature. Resulting from the real-time requirement of 20 kHz or even larger in SiC components and system nonlinearity, even chaotic behavior, offline trained RL control strategies for dc–dc converters may sustain unexpected hurdles during the sim-to-real transfer procedure. As a pioneer work, this article studies a transfer method of reinforcement learning for a dc–dc buck converter, aiming to realize the model-free DRL controller from simulation to practical implementation. To this aim, as briefly depicted in Fig. 1, a duty ratio mapping (DRM) method is proposed for a DRL control structure in the dc–dc converters. First, a model-free reinforcement learning controller based on the Deep Q Learning (DQN) algorithm is adopted to the dc–dc buck converter control with a discrete duty ratio designed as the control actions. Second, the DRM is constructed by the voltage conversion ratio under steady-state conditions to transfer the DRL strategy from the simulation to the real-life environment. Third, an approximated linear function is adopted to reconstruct the DRM by measuring the voltage conversion ratio and output current. Finally, the experimental setup is established to evaluate the performance of the proposed transfer approach. The results indicate that the TL using the proposed DRM strategy provides a successful realization to address the challenges of controlling the dc–dc converter with the presence of both internal uncertainties and external variability. The main contributions of this article can be summarized in the following statements.

1) In order to solve the TL issue for power converters, a new DRM methodology is proposed, which guarantees the realization of DRL in practice.
2) Based on the proposed strategy, we are now able to realize the practical implementation of the DRL approach into a dc–dc buck converter while both transient-time and steady-state control performance can be significantly improved in reference to the existing related results.
3) The adaptability of the control system regarding circuit component parameter variations is evidently verified by the experimental tests, i.e., the same offline trained DRL controller is able to control various buck circuits with a large range of parameter uncertainties.

The rest of this article is organized as follows. Section II describes the basic structure of the dc microgrid with dc–dc converters and the methodology of DRL. In Section III, a new DRM method is proposed to reduce the gap between the simulation and the real dc–dc buck system. Section IV gives the experiments of transfer with the DRM to demonstrate the validity of the proposed approach. Finally, Section V concludes this article.

## II. Problem Formulation and Preliminaries

### A. Problem Formulation

Fig. 2 shows a general layout of a typical dc microgrid. Various dc sources including renewable sources (RESs), energy storage systems (ESs), and the ac grid are connected to the source bus. The dc–dc buck converter is adopted to regulate the output voltage at a nominal value for loads. The loads maintain constant power and are supplied by dc load bus via dc–dc converters, dc–ac inverters, which is able to be classified as constant power loads (CPLs). With the determined nominal voltage, the instantaneous current generated by CPLs can be depicted as: $i_{\text{CPL}} = \frac{P_{\text{CPL}}}{v_o}$.

The minimal conversion unit in the dc microgrid is the dc–dc buck converter shown in Fig. 3 and its average model is given by [29]

$$\begin{cases} \dot{i}_L = \frac{Ed}{L} - \frac{v_o}{L} \\ \dot{v}_o = \frac{i_L}{C} - \frac{v_o}{RC} - \frac{P_{\text{CPL}}}{Cv_o}. \end{cases} \tag{1}$$

*Remark 1:* In engineering applications, the values of $L$ and $C$ may be affected by the external environment. Especially after

the temperature and operating frequency change, their values will deviate from the nominal value due to the constraint of the characteristic curve. On the one hand, these changes affect the accuracy of the model, on the other hand, they may lead to serious nonlinearities of the system to a certain extent.

The control objective of the dc microgrid is to regulate the voltage of the load bus at a nominal value by the dc–dc buck converter. In this article, based on a previous DRL control approach in [30], we are aiming to propose a sim-to-real transfer procedure via a novel DRM strategy. In this regard, both the transient-time and steady-state control performance could be guaranteed. Meanwhile, adaptive capability for system parameter changes could be guaranteed with the proposed control scheme.

### B. DRL Methodology Revisit

Reinforcement learning is a field of machine learning, which researches how to act based on the environment to maximize the expected benefits. Based on whether the agent learns from the environment, RL can be divided into two categories: model-free and model-based. Compared with model-based RL algorithms, model-free RL algorithms converge asymptotically [31]. In other words, the model-free RL algorithms ensure that the agent obtains the optimal solution after numerous interactions with the environment. Thereafter, the DQN algorithm of this article is based on one of the model-free RL algorithms.

DQN is a value-based DRL algorithm, which operates according to the maximum Q value in the environment. The key of DQN is a variant of Q-Learning, which inputs raw data and outputs a value function to evaluate the effectiveness of the current action, thereby training the neural network. In the DQN framework, the approximation of the current Q value $y_j$ is obtained from a deep neural networks (DNN). Each decision will be executed and the Q value will be updated according to the following equation:

$$y_j = \begin{cases} r_j, \text{ if episode terminate at step } j+1 \\ r_j + \gamma \max\limits_{a} \hat{Q}\left(s_{j+1}, a_{j+1}, \theta^-\right), \text{otherwise.} \end{cases} \tag{2}$$

The gradient descent method is used to reduce the root mean square error of the Q value loss function as much as possible to train the parameters [32]. It is depicted as

$$L(\theta) = E[(r + \gamma \max_{a'} Q\left(s', a'; \theta\right) - Q(s, a; \theta))^2]. \tag{3}$$

Fig. 4 shows the brief structure of the DQN algorithm. It splits the neural network into the following two parts:

1) one Q network updates the Q value synchronously;
2) the other target Q network calculates the target Q value $y_j$, and automatically synchronizes the weight of the Q network to the target Q network after a fixed time step.

In the process of interaction between the DQN agent and the environment, the experience data $\{s_t, a_t, r_t, s_{t+1}\}$ obtained at each time step will be saved to the replay memory $D$, and the neural network would be trained by the batch sampling of the experience data. The action at each step is adopted by $\varepsilon$-greedy strategy depicted as

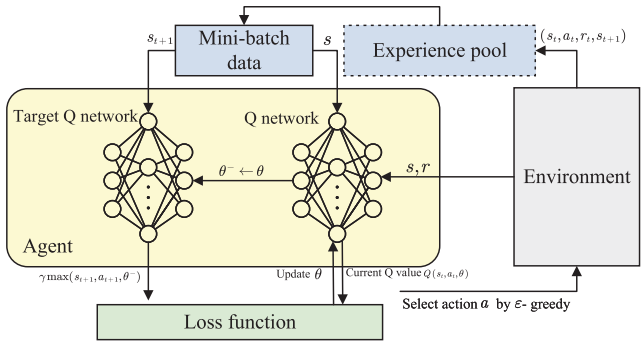


Fig. 4. Brief structure of the DQN strategy.

$$a_t = \begin{cases} \arg\max\limits_{a} Q\left(s_t, a_t\right), & \text{if } p < \varepsilon; \\ a_{t-\text{ran}}, & \text{otherwise.} \end{cases} \tag{4}$$

### C. Sim-to-Real TL Revisit

Transferring the DRL strategy from the simulation to practical implementation is necessary for realizing complex real-world engineering applications with RL-based controllers. However, it is not a specific problem of the DRL algorithm but a general problem of all machine learning (ML). Most DRL algorithms provide end-to-end strategies, which are control mechanisms that receive raw sensor data as input and generate direct activation commands as output. The two dimensions of dc–dc converters can be separated similarly. It is of practical significance for simulators to be more accurate to handle the gap between simulation and implementation. Meanwhile, by acknowledging the presence of unmodeled dynamics, this problem can be much more severe, including the general problems of ML to deal with real-world situations that cannot be considered in the simulation.

Therefore, TL is a strategy that takes advantage of the knowledge learned in the source task to facilitate learning the new task goal. It is mainly utilized to solve the low sampling efficiency and security issues in robotics when the robot or manipulator directly interacts with the environment in the implementation [33]. Meanwhile, by utilizing proper transfer methods, end-to-end strategies such as DRL are able to obtain the capacity to achieve a desirable performance in the environment where limited data has been gathered, or the capacity to obtain a satisfactory performance in a related environment [34].

## III. Transfer From Simulation to Implementation

In this section, inspired by the task mapping method, a TL method is adopted to realize the reflection of the action-value function for a DRL controller in the real-life environment, as shown in Fig. 5. Detailed design procedures are given as follows.

### A. DRL Controller Design

In a previous work [30], the authors have proposed a DQN algorithm based on superior learning to adjust the duty ratio for the dc–dc buck converter. The control diagram of the proposed

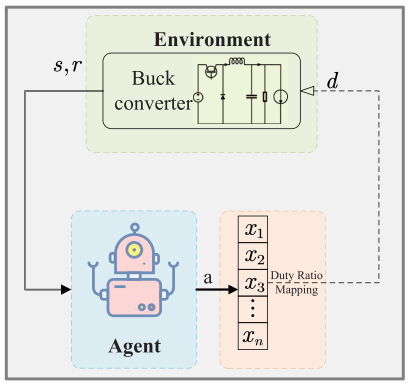


Fig. 5. Overview of transferring the DRL method into real-life systems.

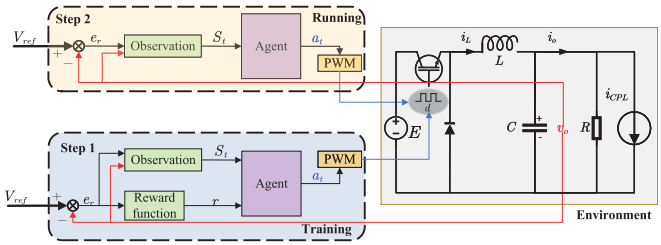


Fig. 6. Control diagram of the DRL controller.

DRL controller is presented in Fig. 6. It mainly includes the following two stages.

1) *Step 1:* The agent is trained offline in the MATLAB RL environment, in this stage, it cannot obtain the desired control performance initially. After several episodes interacting with the environment, the DNN is optimized automatically by the DQN strategy, which is able to produce the duty of the pulsewidth modulation (PWM) according to the signals from the environment.
2) *Step 2:* Aiming to extract the trained agent, we reorganized the DNN with a MATLAB function in which the weight and bias are copied from the agent in Step 1. Then, the end-to-end strategy is completed finally and if the environment gives the signals to this strategy function, the module will generate the corresponding action.

The design of state space, action space, reward/penalty function, and exploration strategy is illustrated in the following steps.

***1) State Space:*** The output voltage $v_o$ and the tracking error $e(t) = v_o(t) - V_{\text{ref}}$ are considered as the basic signals to obtain the system state. The state is depicted as: $S_t = \{v_o(t), v_{o\_\text{del}}(t), \frac{\text{d}v_o(t)}{\text{d}t}, e(t), e_{\text{del}}(t), \frac{\text{d}e(t)}{\text{d}t}\}$.

***2) Action Space:*** The switch control is chosen as a reference to design a discrete action space. The approximate range of the duty ratio is determined by the ratio of the reference voltage to the actual input voltage. Three variables are defined: the steady-state value $\xi$, the fluctuation range $\phi$, and minimum change interval $c$. Thereby, a discrete action space is constructed as $A \in [\xi - \phi,\ \xi + \phi]$.

***3) Reward Function With Subgoals:*** The reward function is designed by the tracking error $e(t)$ between the current state and control objective. Two subgoals are utilized to guide the learning agent, i.e., $\epsilon_1$ and $\epsilon_2$. $\beta_1$, $\beta_2$, and $\beta_3$ are selected as the reward/penalty coefficients. The reward function of the proposed controller is shown as follows:

$$r = \begin{cases} \beta_1 - \beta_3 e(t), & \text{if } 0 \le |e(t)| < \epsilon_1 \\ \beta_2 - \beta_3 e(t), & \text{if } \epsilon_1 \le |e(t)| \le \epsilon_2 \\ -\beta_3 e(t), & \text{else.} \end{cases} \tag{5}$$

***4) DNN Design:*** The network has seven layers, including an input layer, three fully connected layers, two hidden layers, and an output layer. The hidden layers have both 64 neurons. The activation function of each hidden layer uses the Relu function.

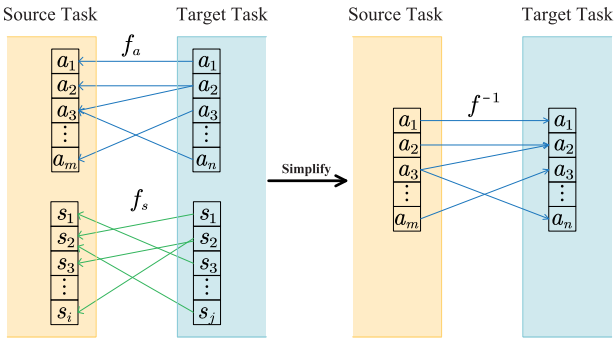


Fig. 7. Simplification of the DRM strategy.

## B. Duty Ratio Transfer Functional

In what follows, a DRM method is introduced in detail as the task mapping construction and the mapping function approximation regarding the DRL control issue for a dc–dc buck converter.

It is well known that the model in the simulation environment of a dc–dc converter behaves with a large deviation from the actual circuit in the real environment. The DRL control strategy stabilizes the dc power systems to reach the steady-state value with a high control performance. The task mapping can be constructed by the parameters of the converters under steady-state conditions, ignoring the dynamic behaviors. Thereafter, the output voltage and the voltage deviation of the dc–dc buck converter conform to the following formulas:

$$\begin{cases} v_{i,\text{real}} = v_{i,\text{sim}} \\ v_{o,\text{real}} = v_{o,\text{sim}} = v_{\text{ref}} \\ e_{o,\text{real}} = e_{o,\text{sim}} = v_o(t) - v_{\text{ref}} = 0. \end{cases} \tag{6}$$

For the sake of simplicity, as shown in Fig. 7, we only need to consider the state-action transform relationship between states $v_o$ and actions $d$.

On the one hand, in the simulation environment, the steady-state voltage conversion ratio of the buck converter is equal to the duty ratio, that is,

$$v_{o,\text{sim}} = d_{\text{sim}} v_{i,\text{sim}}. \tag{7}$$

On the other hand, the action-state transformation relationship in the real-life environment can be expressed as

$$v_{o,\text{real}} = f(d_{\text{real}}, v_{o,\text{real}}, i_{o,\text{real}})v_{i,\text{real}} \tag{8}$$

where $f$ is a monotone mapping between the duty ratio in the simulation and real-life environment.

According to relations (6)–(8), the task mapping of the action-state transform can be converted to a DRM

$$d_{\text{sim}} = f(d_{\text{real}}, v_{o,\text{real}}, i_{o,\text{real}}) \tag{9}$$

i.e.,

$$d_{\text{real}} = f^{-1}(d_{\text{sim}}, v_{o,\text{real}}, i_{o,\text{real}}). \tag{10}$$

Thus, a learned DRL control strategy in the simulation environment can be transferred to the real-life environment by creating a DRM given the dc–dc converter simulation environment $D_{\text{sim}} = (S_{\text{sim}}, A_{\text{sim}}, P_{\text{sim}}, R_{\text{sim}})$ and the real-life environment $D_{\text{real}} = (S_{\text{real}}, A_{\text{real}}, P_{\text{real}}, R_{\text{real}})$.

### C. Mapping Function Approximation

In real-life experiments, it is necessary to obtain the form of the function approximation for the DRM. Hence, a simple linear function is adopted to approximate the relationship. According to (10), the DRM is a mapping between the actual duty ratio, simulated duty ratio, and output current under steady-state conditions. Therefore, the sampled data of the DRM can be obtained through the experimental results in the simulation environment and the real-life environment under steady-state operating conditions. Assuming that the actual duty ratio is a linear function of the simulated duty cycle and output current, the approximate function can be regressed by a two-degree linear approximation using a partial least-square method with a set of sampled data.

Referring to Algorithm 1, the detailed process to approximate the DRM is given as follows.

1) Given $d_{\text{real},k} \in A$ in the action set and $P_{o,k} \in [P_{o,\min}, P_{o,\max}]$ in the constant power load range.
2) Given the duty ratio action $d_{\text{real},k} \in A$ and the output power $P_{o,k}$ in the simulation environment, the simulation environment information $D_{\text{sim},k}(v_{o,\text{sim},k}, d_{\text{real},k}, i_{o,\text{sim},k})$ with the output voltage and current can be obtained under steady-state operation.
3) Given the output power $P_{o,k}$ in the real environment, tune the real duty ratio $d_{\text{real},k}$ to the simulation duty ratio action $d_{\text{sim},k}$ under steady-state operating condition.
4) The output current $i_{o,\text{real},k}$ and voltage $v_{o,\text{real},k}$ of the dc–dc converter of the real environment information $D_{\text{real},k}(v_{o,\text{real},k}, d_{\text{real},k}, i_{o,\text{real},k})$ can be measured.
5) The simulation duty ratio $d_{\text{sim},k}$ corresponds to real duty ratio $d_{\text{real},k}$, which is calculated by (7), i.e., $d_{\text{sim},k} = v_{o,\text{real},k}/v_{i,\text{sim},k}$.
6) Repeat the procedures from 1) to 5). A set of sampled data $S(v_{o,k}, d_{\text{real},k}, i_{o,k})$ can be obtained through the experimental results.

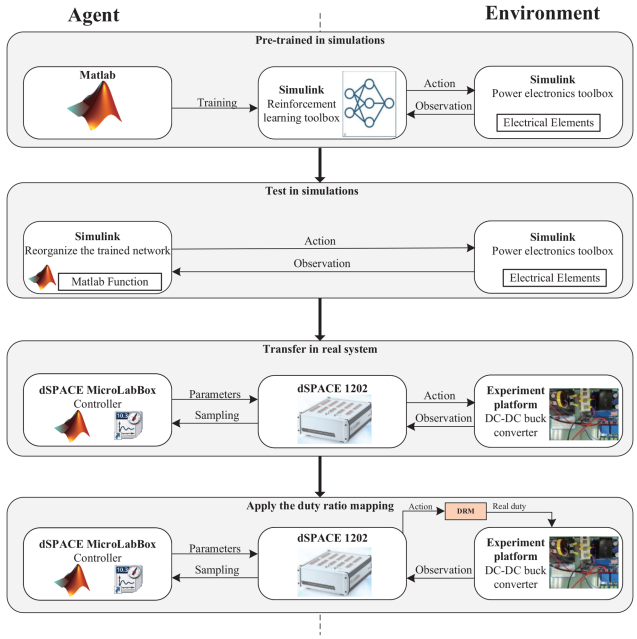


Fig. 8. Proposed sim-to-real procedure.

**Algorithm 1:** Acquisition of the DRM.

**Data:** Output power $P_o$, output current $i_{o,\text{real}}$, output voltage $v_{o,\text{real}}$, simulation duty ratio action $d_{\text{sim}}$
**Result:** Approximate function $d_{\text{real}} = f(d_{\text{sim}})$

```
Initialize the experimental equipment;
Determine the range of P_o, d_real;
Initialize the information table D_sim and D_real;
k ← 0;
while P_{o,k} ∈ [P_{o,min}, P_{o,max}] and d_{real,k} ∈ A do
    Obtain the corresponding information of simulation
     according to d_{real,k};
    Keep the system stable under the given condition
     P_{o,k};
    Obtain the output voltage v_{o,real,k} and the output
     current i_{o,real,k} under certain working condition;
    Calculate the parameters of the actual system
     d_{real,k};
    Update information table D_{sim,k} and D_{real,k};
    Pause equipment and cool down;
    k ← k + 1;
end
Obtain the mapping relationship d_real by the least-square
 method.
```

7) The approximate function $d_{\text{real}} = ad_{\text{sim}} + bi_{o,\text{real}} + c$ of the DRM is regressed by the partial least-square method through multiple groups of transfer sampled data.

### D. Sim-to-Real Procedure

The development of the proposed controller can be split into four steps, as depicted in Fig. 8. First, the power electronics toolbox is utilized to simulate the actual dc–dc converter system. The RL toolbox acts as an intermediary to interact with the

TABLE II
PARAMETERS OF THE BUCK CONVERTER

| Variables | Description | Value |
|---|---|---|
| $E$ | Input voltage | 200V |
| $V_{\text{ref}}$ | Bus voltage | 100V |
| $L$ | Inductance | 1/2/6.8mH |
| $C$ | Capacitance | 1/1/1.8mF |
| $f$ | Switching frequency | 20kHz |
| $k_{vp}, k_{vi}$ | PI gains of voltage control loop | 4.5, 50 |
| $k_{vp}, k_{vi}$ | PI gains of current control loop | 0.02, 30 |

TABLE III
DQN LEARNING PARAMETERS

| Variables | Description | Value |
|---|---|---|
| $\alpha$ | Learning rate | 0.001 |
| $\gamma$ | Discount factor | 0.95 |
| $B$ | Replay memory capacity | 2e-6 |
| $b$ | Minibatch size | 256 |
| $\varepsilon$ | Exploration rate | 0.1 |
| $A$ | Action space | $[0.45,\ 0.55]$ |
| $c$ | Minimum change interval | 0.01 |
| $\beta_1, \beta_2$ | Reward coefficients | 10, 1 |
| $\beta_3$ | Penalty coefficient | -10 |
| $\epsilon_1, \epsilon_2$ | Sub-goals of the reward function | 0.1, 1 |
| $M$ | Number of neurons | 64 |
| $N$ | Number of neurons | 64 |

environment, and the RL agent's weights are pretrained in the simulation. Second, the weights are reorganized to synthesize a new network with a MATLAB function, which can be regarded as a black box to generate the PWM duty ratio by observation. Third, the compiler exported the controller to C code and imported them to a dSPACE MicroLabBox with a real-time kernel automatically. Then, the controller is running in the real-life system without DRM. Finally, the DRM is added to the DRL controller to handle the gap between the simulation and the experimental platform.

## IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

The detailed offline training parameters of the dc–dc buck converter are shown in Table II. The hyperparameters for the design of the DQN controller are depicted in Table III. In the training process, the initial state of the CPL is set as 200 W. Later on, it switches to a new working condition at 0.14 s and drops back to 200 W at 0.2 s. The CPL is switched to 200, 500, and 800 W, respectively.

### A. Experiment Setup

The experimental setup depicted in Fig. 9 is built to verify the effectiveness of the proposed transfer strategy, which consists of a custom-designed dc power supply (Chroma 62012P-600-8), a custom-designed dc electronic load (Chroma 63202E-150-200), a dc–dc buck converter, and dSPACE 1202. The control algorithm is embedded in dSPACE to generate PWM signals for

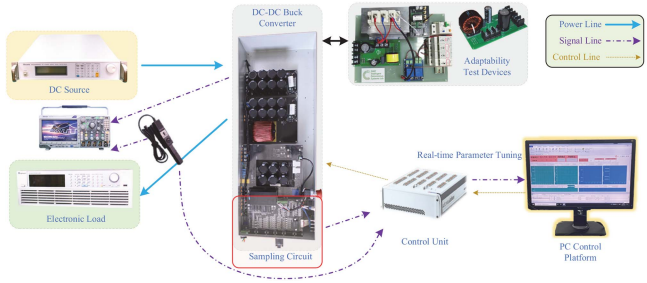


Fig. 9. Experiment setup.

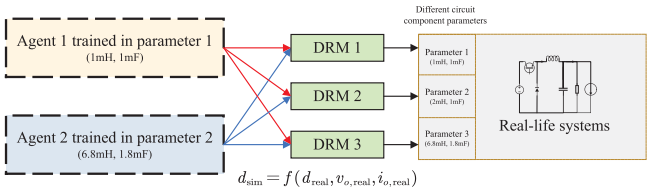


Fig. 10. Experimental operation of *Case II*.

the dc–dc buck converter with a switching frequency of 20 kHz. The dc electronic load is configured to operate in constant power mode to simulate the CPL. The detailed nominal parameters of the real-time implementation are consistent with the offline training environment.

### B. Experimental Results

***1) Transferring Experiments:*** According to the procedure depicted in Fig. 8, one can reorganize the pretrained network into a MATLAB function. When building the model in the Controldesk (host software of dSPACE), one can use this function as the controller, and use the DRM function to compensate for real-time deviations. By clicking on the appropriate block, any setting or change of parameters can be achieved. In this test workbench, the DRM is obtained based on about 300 sets of data tested, and the final expression relationship is $a = 0.9979$, $b = -0.0002332$, and $c = 0.06103$. The feasibility of the DRM-based DRL controller with respect to the stable ideal CPLs is studied in this section. The power of CPL is set as a constant value of $P = 200$, 400, 600, and 800 W, respectively. In the initial state, the power of CPL is set as 200 W, and at a certain moment, it is changed by controlling the block in the host computer. With the concerned case, the experimental outcomes of bus voltage and CPL's current, respectively, for the proposed controller are depicted in Fig. 11. In summary, the steady-state error of the voltage with the DRM is reduced significantly and the voltage fluctuation decreases compared to the situation without the DRM, especially as the working conditions become larger. Meanwhile, the controller employing the DRM has shown a strong ability to attenuate the fluctuation of the output voltage.

***2) Experiments With Different Circuit Component Parameters:*** In order to ensure the validity of the proposed method and test that the method may be applied to different

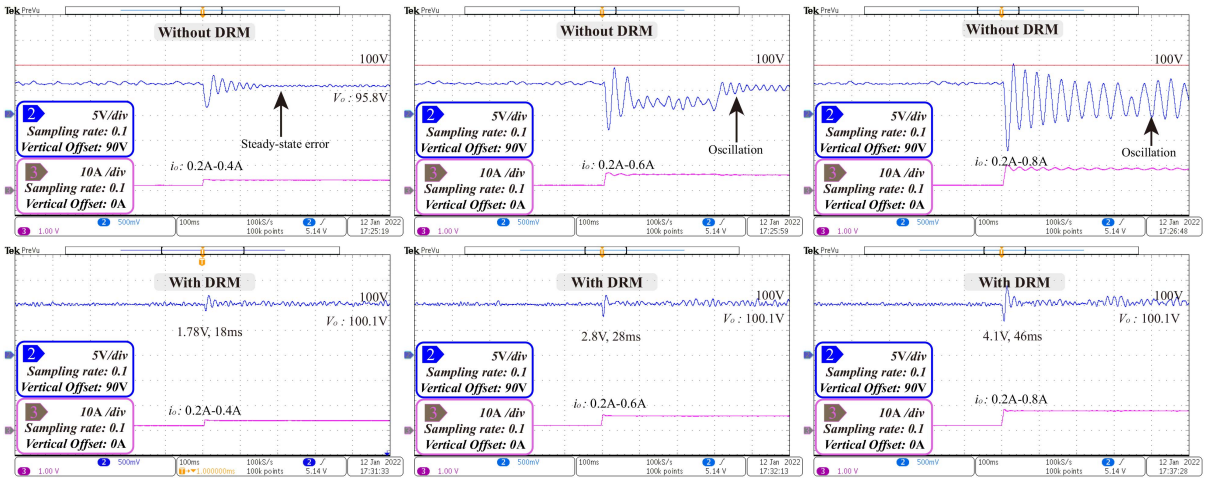


Fig. 11. Voltage/current response curves with and without DRM in different working conditions (200–400 W, 200–600 W, and 200–800 W).

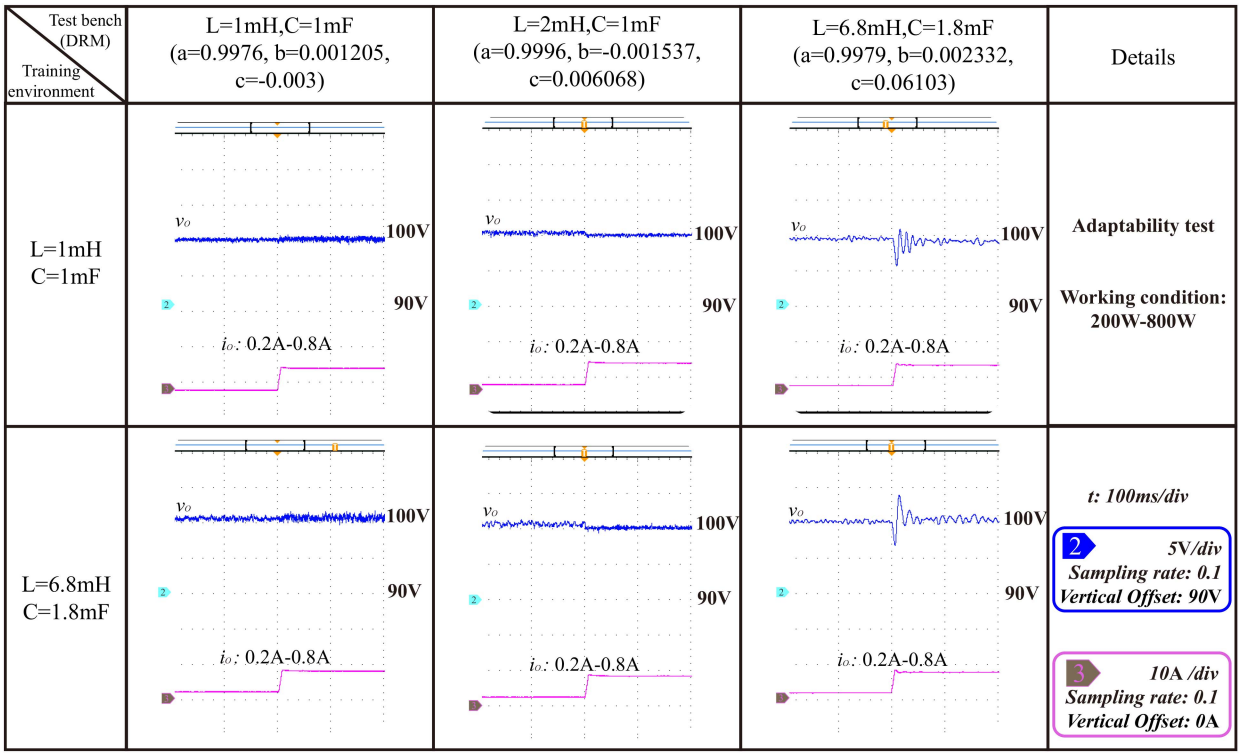


Fig. 12. Adaptability test of the proposed sim-to-real procedure-based DRL control methodology.

converter devices, we separately train models for two pairs of different circuit parameters ($L = 1$ mH, $C = 1$ mF and $L = 6.8$ mH, $C = 1.8$ mF) and conduct experiments with different buck converters ($L = 1$ mH, $C = 1$ mF, $L = 2$ mH, $C = 1$ mF, and $L = 6.8$ mH, $C = 1.8$ mF). The working condition is set as CPL variation from 200 to 800 W. Fig. 10 shows the specific experimental process in this case.

Fig. 12 illustrates how the proposed controller stabilizes the buck converter under the reference voltage variations. Obviously, the agents trained in different training environments show similar experimental results by utilizing the proposed DRM methodology. Subject to different training parameters, there is a slight deviation in transient performance including bus voltage and CPL current with the circuit parameters set as $L = 6.8$ mH and $C = 1.8$ mF.

***3) Performance Comparison With the PI Controller:*** The comparisons of the transient-time performance of the double-loop PI controller and the DRL controller are shown in Fig. 13. Circuit parameters $L = 1$mH and $C = 1$ mF are selected as the test bench. Several error measurement criteria including integral absolute error and integral square error are compared by bar chart in Fig. 14. It is evidently observed that the implementation of the proposed DRL controller has outperformed the classical PI controller in transient-time control performance indexes. This

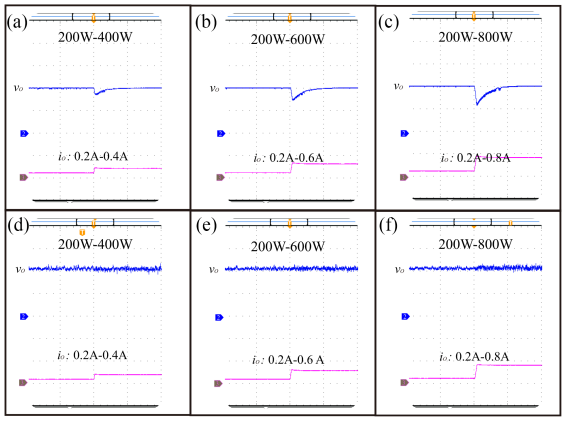


Fig. 13. Transient-time performance comparisons of the (a)–(c) PI controller and (d)–(f) DRL controller.

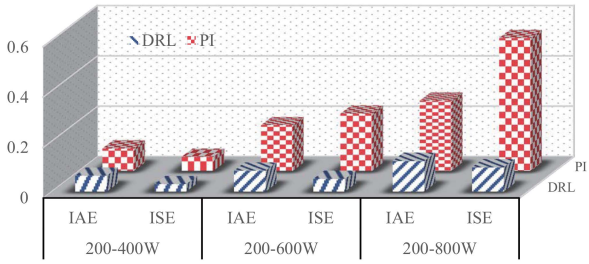


Fig. 14. Performance indexes of the DRL controller and PI controller.

TABLE IV
COMPARISON OF SINGLE-STEP OPERATION TIME OF DIFFERENT CONTROLLERS

| Controller | Total time/total calls | Time of each control cycle |
|---|---|---|
| PI controller | 17.307/159367438 | $0.1086\mu s$ |
| DRL controller | 70.14/156814415 | $0.4473\mu s$ |
| MPC | 350.974/156743072 | $2.2392\mu s$ |

fact could bring much confidence for power engineers to apply the DRL controllers in certain practice.

*Remark 2:* In order to verify the feasibility of the proposed method, the comparison results of execution time under different controllers are listed in Table IV. The model predictive controller is designed by MATLAB toolbox in which a disturbance observer is used to compensate for the steady-state error [35]. The simulation is carried out on an AMD Ryzen 5 4600 G 3.7 GHz, 8-cores, 8-GB RAM, MATLAB/Simulink 2021a environment. It is shown that compared with the MPC method, DRL overcomes the issue of long online computation times by precomputing the optimal solutions offline, a concept similar to parametric programming in explicit MPC. Therefore, the online computation burden of DRL is not a main concern for engineers, which makes its application less demanding on the performance of the processors.

## V. CONCLUSION

In this article, the transfer of the DRL controller from offline training to implementation was proposed. A DRM method was proposed to transfer a model-free DRL-based dc–dc buck controller from simulation environment to the real world via the voltage conversion ratio. A simple linear function was adopted to approximate the mapping by the measurement data stream. The presented DRL-based controller was implemented on a typical laboratory hardware system in real time with a frequency of 20 kHz, which was much larger than that in other fields such as robotics. On one hand, owing to the fact that the proposed DRM methodology had a strong potential to improve the optimization effect of the dc–dc buck converter, future works can be extended to motion control and microgrid applications, etc. On the other hand, considering that the DQN algorithm was discrete and may cause the problems of limited control accuracy, continuous interval learning methods can be investigated in future work. In addition, other transfer methods considering system dynamics and the generalizability of DRL in power electronics will be further studied in future works to support more practical implementations.

## REFERENCES

[1] G. Li and B. Zhang, "A novel weak signal detection method via chaotic synchronization using Chua's circuit," *IEEE Trans. Ind. Electron.*, vol. 64, no. 3, pp. 2255–2265, Mar. 2017.
[2] N. Bottrell, M. Prodanovic, and T. C. Green, "Dynamic stability of a microgrid with an active load," *IEEE Trans. Power Electron.*, vol. 28, no. 11, pp. 5107–5119, Nov. 2013.
[3] Y. Zhang, S. Wang, and Y. Chu, "Investigation of radiated electromagnetic interference for an isolated high-frequency DC-DC power converter with power cables," *IEEE Trans. Power Electron.*, vol. 34, no. 10, pp. 9632–9643, Oct. 2019.
[4] Y. Yuan, C. Chang, Z. Zhou, X. Huang, and Y. Xu, "Design of a single-input fuzzy PID controller based on genetic optimization scheme for DC-DC buck converter," in *Proc. Int. Symp. Next- Gener. Electron.*, 2015, pp. 1–4.
[5] Q. Xu, Y. Yan, C. Zhang, T. Dragicevic, and F. Blaabjerg, "An offset-free composite model predictive control strategy for DC/DC buck converter feeding constant power loads," *IEEE Trans. Power Electron.*, vol. 35, no. 5, pp. 5331–5342, May 2020.
[6] 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.
[7] C. Zhang, X. Wang, P. Lin, P. X. Liu, Y. Yan, and J. Yang, "Finite-time feedforward decoupling and precise decentralized control for DC microgrids towards large-signal stability," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 391–402, Jan. 2020.
[8] X. Li, X. Zhang, W. Jiang, J. Wang, P. Wang, and X. Wu, "A novel assorted nonlinear stabilizer for DC-DC multilevel boost converter with constant power load in DC microgrid," *IEEE Trans. Power Electron.*, vol. 35, no. 10, pp. 11181–11192, Oct. 2020.
[9] F. Li et al., "Review of real-time simulation of power electronics," *J. Modern Power Syst. Clean Energy*, vol. 8, no. 4, pp. 796–808, 2020.
[10] S. Kapat and P. T. Krein, "A tutorial and review discussion of modulation, control and tuning of high-performance DC-DC converters based on small-signal and large-signal approaches," *IEEE Open J. Power Electron.*, vol. 1, pp. 339–371, Sep. 2020.
[11] P. Chaudhary and M. Rizwan, "Voltage regulation mitigation techniques in distribution system with high PV penetration: A review," *Renewable Sustain. Energy Rev.*, vol. 82, pp. 3279–3287, 2018.
[12] M. Adibi and J. van der Woude, "A reinforcement learning approach for frequency control of inverted-based microgrids," *IFAC-PapersOnLine*, vol. 52, no. 4, pp. 111–116, 2019.
[13] S. Wang et al., "A data-driven multi-agent autonomous voltage control framework using deep reinforcement learning," *IEEE Trans. Power Syst.*, vol. 35, no. 6, pp. 4644–4654, Nov. 2020.
[14] M. H. Khooban and M. Gheisarnejad, "A novel deep reinforcement learning controller based type-II fuzzy system: Frequency regulation in microgrids," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 5, no. 4, pp. 689–699, Aug. 2021.
[15] S. Zhao, F. Blaabjerg, and H. Wang, "An overview of artificial intelligence applications for power electronics," *IEEE Trans. Power Electron.*, vol. 36, no. 4, pp. 4633–4658, Sep. 2021.

[16] 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.
[17] Y. Xia, Y. Xu, Y. Wang, and S. Dasgupta, "A distributed control in islanded DC microgrid based on multi-agent deep reinforcement learning," in *Proc. 46th Annu. Conf. IEEE Ind. Electron. Soc.*, 2020, pp. 2359–2363.
[18] M. H. Khooban and M. Gheisarnejad, "A novel deep reinforcement learning controller based type-II fuzzy system: Frequency regulation in microgrids," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 5, no. 4, pp. 689–699, Aug. 2021.
[19] M. Gheisarnejad and M. H. Khooban, "IoT-based DC/DC deep learning power converter control: Real-time implementation," *IEEE Trans. Power Electron.*, vol. 35, no. 12, pp. 13621–13630, Dec. 2020.
[20] S. Richter, C. N. Jones, and M. Morari, "Computational complexity certification for real-time MPC with input constraints based on the fast gradient method," *IEEE Trans. Autom. Control*, vol. 57, no. 6, pp. 1391–1403, Jun. 2012.
[21] A. Visioli, *Practical PID Control*, ser. Advances in Industrial Control. London, U.K.: Springer, 2006.
[22] 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.
[23] S. Borreggine, V. G. Monopoli, G. Rizzello, D. Naso, F. Cupertino, and R. Consoletti, "A review on model predictive control and its applications in power electronics," in *Proc. AEIT Int. Conf. Elect. Electron. Technol. Automot.*, 2019, pp. 1–6.
[24] R. Nian, J. Liu, and B. Huang, "A review on reinforcement learning: Introduction and applications in industrial process control," *Comput. Chem. Eng.*, vol. 139, 2020, Art. no. 106886.
[25] Z. Zhu, K. Lin, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," 2020, *arXiv:2009.07888*.
[26] F. L. Da Silva and A. H. R. Costa, "A survey on transfer learning for multiagent reinforcement learning systems," *J. Artif. Intell. Res.*, vol. 64, pp. 645–703, 2019.
[27] G. Book et al., "Transferring online reinforcement learning for electric motor control from simulation to real-world experiments," *IEEE Open J. Power Electron.*, vol. 2, pp. 187–201, Apr. 2021.
[28] F. Muratore, F. Treede, M. Gienger, and J. Peters, "Domain randomization for simulation-based policy optimization with transferability assessment," in *Proc. Conf. Robot Learn.*, 2018, pp. 700–713.
[29] P. Lin, W. Jiang, J. Wang, D. Shi, C. Zhang, and P. Wang, "Toward large signal stabilization of floating dual boost converter powered DC microgrids feeding constant power loads," *IEEE Trans. Emerg. Sel. Topics Power Electron.*, vol. 9, no. 1, pp. 580–589, Feb. 2021.
[30] 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.
[31] D. Ormoneit and Ś. Sen, "Kernel-based reinforcement learning," *Mach. Learn.*, vol. 49, no. 2, pp. 161–178, 2002.
[32] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[33] T.-H. Pham, G. De Magistris, and R. Tachibana, "Optlayer-practical constrained optimization for deep reinforcement learning in the real world," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 6236–6243.
[34] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *Found. Trends Mach. Learn.*, vol. 11, no. 3–4, pp. 219–354, 2018.
[35] S. Li, J. Yang, W.-H. Chen, and X. Chen, *Disturbance Observer-Based Control: Methods and Applications*. Boca Raton, FL, USA: CRC Press, 2014.

**Chenggang Cui** (Member, IEEE) received the B.E. degree in automation engineering from Jilin University, Changchun, China, in 2004, and the Ph.D. degree in control theory and control from Zhejiang University, Hangzhou, China, in 2010.

He worked with Shanghai Institute for Advanced Studies, Chinese Academy of Sciences, involved in energy management and optimal scheduling from 2012 to 2015. He has been with the School of Automation, Shanghai University of Electric Power, where he is currently an Associate Professor. His research interests include the control and schedule of renewable energy systems and microgrid.

**Tianxiao Yang** (Student Member, IEEE) received the B.E. degree in electrical engineering from the Shenyang University of Technology, Shenyang, China, in 2020. He is currently working toward the master's degree in automatic control with the Shanghai University of Electric Power, Shanghai, China.

His research interests include deep reinforcement learning theory and its applications for microgrid and power systems.

**Yuxuan Dai** received the B.E. degree in electrical engineering from the Jiangsu University of Science and Technology, Zhenjiang, China, in 2019. He is currently working toward the master's degree in automatic control with the Shanghai University of Electric Power, Shanghai, China.

His main research interests include deep reinforcement learning with applications to dc–dc converter and microgrid systems.

**Chuanlin Zhang** (Senior Member, IEEE) received the B.S. degree in mathematics and the Ph.D. degree in control theory and control engineering from the School of Automation, Southeast University, Nanjing, China, in 2008 and 2014, respectively.

He was a Visiting Ph.D. Student with the Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX, USA, from 2011 to 2012; a Visiting Scholar with the Energy Research Institute, Nanyang Technological University, Singapore, from 2016 to 2017; and a Visiting Scholar with Advanced Robotics Center, National University of Singapore, from 2017 to 2018. Since 2014, he has been with the College of Automation Engineering, Shanghai University of Electric Power, Shanghai, China, where he is currently a Professor. His research interests include nonlinear system control theory and applications for power systems.

**Qianwen Xu** (Member, IEEE) received the B.Sc. degree in electrical engineering from Tianjin University, Tianjin, China, in 2014, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2018.

She worked as a Postdoctoral Research Fellow with Aalborg University, Aalborg, Denmark, and a Wallenberg-NTU Presidential Postdoc Fellow with Nanyang Technological University, Singapore, during 2018–2020. She was also a Visiting Researcher with Imperial College London during March 2020 to June 2020. She is currently an Assistant Professor with the Department of Electric Power and Energy Systems, KTH Royal Institute of Technology, Stockholm, Sweden. Her research interests include advanced control, optimization, and AI application for microgrid and smart grid.

Dr. Xu serves as the Vice Chair in IEEE Power and Energy Society and Power Electronics Society, Sweden Chapter, and an Associate Editor for IEEE TRANSACTIONS ON SMART GRID and IEEE JOURNAL OF EMERGING AND SELECTED TOPICS IN POWER ELECTRONICS. She was also the recipient of Humboldt Research Fellowship, Excellent Doctorate Research Work from Nanyang Technological University, Best paper award in IEEE International Symposium on Power Electronics for Distributed Generation Systems (PEDG) 2020, etc.