Journal of Systems Engineering and Electronics
Vol. 34, No. 1, February 2023, pp.117 – 128

Reinforcement learning-based scheduling of multi-battery energy storage system

CHENG Guangran [1,2], DONG Lu [3], YUAN Xin [1], and SUN Changyin [1,2,*]

1. School of Automation, Southeast University, Nanjing 210096, China; 2. Peng Cheng Laboratory, Shenzhen 518066, China;
3. School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China

Abstract: In this paper, a reinforcement learning-based multi-battery energy storage system (MBESS) scheduling policy is proposed to minimize the consumers' electricity cost. The MBESS scheduling problem is modeled as a Markov decision process (MDP) with unknown transition probability. However, the optimal value function is time-dependent and difficult to obtain because of the periodicity of the electricity price and residential load. Therefore, a series of time-independent action-value functions are proposed to describe every period of a day. To approximate every action-value function, a corresponding critic network is established, which is cascaded with other critic networks according to the time sequence. Then, the continuous management strategy is obtained from the related action network. Moreover, a two-stage learning protocol including offline and online learning stages is provided for detailed implementation in real-time battery management. Numerical experimental examples are given to demonstrate the effectiveness of the developed algorithm.

Keywords: multi-battery energy storage system (MBESS), reinforcement learning, periodic value iteration, data-driven.

DOI: 10.23919/JSEE.2023.000036

1. Introduction

The cost of electric power in residential environments undergoes distinct variations because of daily fluctuations in load demand and generator capacity [1]. Consumers usually pay little attention to these variations when the electricity price is fixed. Dynamic pricing makes it possible for consumers to reduce high power costs by exploiting price fluctuations [2]. For the past few years, the evolution of microgrids has provided various benefits for the smart grid economy [3]. Particularly with the development of multi-battery energy storage system (MBESS), consumers are able to manage energy storage equipment to reduce charging costs without the need to change their consumption habits. For example, consumers can charge the batteries from the grid when the electricity price is low and release energy when the price is high.

However, it is challenging to efficiently manage MBESS charging/discharging because (i) different characteristics of batteries, extensive consumer behavior patterns, and residential uncertainties make the modeling task not universal, (ii) the battery levels must be kept in their safe zones to prolong their lifetime as far as possible, and (iii) fluctuant load demand and electricity price make the system model time-varying, which makes the management of MBESS a non-stationary optimization problem.

Numerous model-based algorithms have been widely applied to deal with this problem [4−11]. For instance, in [4] and [5], the MBESS optimization problem was converted into a mixed-integer linear programming (MILP) problem, and MILP algorithms were applied to decrease the scheduling cost of the microgrid by reshaping the load, such as peak shaving and valley filling. Some practical operation limitations for battery charging and discharging were considered. In [6], a power generation-side management policy for a modular energy system defined as a general MILP was proposed to promote self-consumption based on a day-ahead market. However, MILP algorithms rely on deterministic rules and abstract consumer-defined models, which may be deviated from the actual MBESS model. Furthermore, MILP-based optimization algorithms are centralized, limited, and lacking in extensibility, as the computational complexity increases with the number of variables in large-scale MBESS [12]. In [8−11], several demand-side storage management strategies based on model predictive control (MPC) were developed to maximize energy power utilization and benefit consumers. The uncertainties about unpredictable customer behavior and time-varying load

Manuscript received December 29, 2021.
*Corresponding author.
This work was supported by the National Key R&D Program of China (2018AAA0101400), the National Natural Science Foundation of China (61921004;62173251;U1713209;62236002), the Fundamental Research Funds for the Central Universities, and Guangdong Provincial Key Laboratory of Intelligent Decision and Cooperative Control.

demand were all taken into account. Although the above MPC-based optimization algorithms succeed in dispatch scheduling, such model-based approaches are limited in complex systems. Since the heterogeneity of ESS makes the modeling task challenging and model-based approaches are difficult to transfer from one scenario to another.

In recent years, the newly developed reinforcement learning (RL) algorithms have been a powerful tool to tackle Markov decision process (MDP) problems in complex decision-making applications [13−21]. Through RL algorithms, agents can directly learn policies by interacting with the system without requiring the system dynamics. In [22], a Q-learning-based approach was proposed for a single electric vehicle (EV) scheduling problem, and a representation network was adopted to predict future electricity prices. In [23], the optimal charging/discharging strategy was learned based on deep Q-learning for single-battery considering an accurate degradation model. A long short term memory (LSTM) model was applied to predict the electricity price for the next day. For MBESS, Q-learning-based algorithms are limited because the discrete action space increases exponentially with the increasing number of control variables. To deal with this problem, some model-free energy optimization works learn the continuous battery power discharge/charging policy. For example, in [24] and [25], smart home energy management algorithms based on deterministic policy gradients were proposed to keep the building in a comfortable temperature range and encourage the consumers to use electricity more efficiently. In [26] and [27], with the aim of power flow optimization and the state of charge (SOC) management, control strategies for MBESS based on deep RL (DRL) were developed.

In practice, agents usually need to respond to time-varying load demand and dynamic electricity price in real-time scheduling, and hence the system dynamics or the state transition probability of the environment changes over time. In [28] and [29], two time-based reinforcement learning (TBRL) algorithms were proposed for residential ESS control. Similarly, in [30], a TBRL-based energy management algorithm for smart power buildings was developed, where Photovoltaic (PV) energy and EVs were taken into account. In [24] and [25], the smart home energy management problem was formulated as time-dependent MDPs, which contains the time index in the system state to obtain the electricity price or the power demand. However, for TBRL algorithms, the optimal function is time-dependent and difficult to obtain when agents interact with time-varying or so-called non-stationary environments. The effectiveness of these TBRL algorithms is usually verified through experiments. The properties of time are different from space. The latter is more reachable and controllable than the former. An object can be moved freely in space, but there are more restrictions in time because time is asymmetric and unidirectional [31]. In this case, the convergence property of such TBRL algorithms has not been analyzed and may diverge, which greatly limits the applications of the TBRL algorithms [32,33]. To tackle this issue, a time-independent RL approach with convergence guarantee is developed in this work for time-varying MBESS charging/discharging problems.

In this paper, we first model the MBESS management problem as an MDP with time-varying transition probability from the consumers' perspective. Our objective is to find an optimal strategy for the battery system scheduling problem without changing residents' consumption habits. Then, we remove the time index from the original MDP and introduce the periodic action-value functions. It is proven that these functions can converge to the optimal functions through periodic value iteration. To save costs in real-time management, we develop a novel model-free algorithm based on the periodic action-value function and deterministic policy gradient (DPG). Finally, the experiment results demonstrate the proposed algorithm can efficiently manage the battery system in the constrained control zone and save costs for consumers. The main contributions of this paper are listed as follows:

(i) Different from the time-dependent MDPs formulated in [24] and [25], we formulate the MBESS charging/discharging scheduling problem as a time-independent MDP. We propose a series of time-invariant action-value functions to describe every time period of a day and introduce the periodic value iteration to learn the optimal action-value functions. The convergence property of the iteration process is guaranteed. A new algorithm called multi-agent deep DPG with incremental number of agents (MADDPG-INA) is proposed to accelerate the learning process from the source MDP with N agents to the target MDP with $N+M$ agents.

(ii) We propose a two-stage RL-based scheduling algorithm, the DPG with periodic action-value functions. After the offline training with the proposed periodic deterministic policy gradient (PDPG) method, the continuous policy can be employed in real-time scheduling in the online stage. We verify by two case studies the effectiveness of the proposed algorithm which shows a higher success rate of convergence than general TBRL algorithms.

The rest of this paper is organized as follows. The system model and the formulated MDP are presented in Section 2. The detailed periodic action-value iteration and a novel PDPG algorithm are presented in Section 3. The numerical experimental results that show the performance of the proposed method are given in Section 4. The conclusion is presented in Section 5.

2. System model and problem formulation

The smart residential MBESS considered in this paper is composed of N batteries with different characteristics. The energy scheduling problem is formulated as an MDP from the perspective of consumers in discrete time steps of 1h. At time step t, we obtain the battery energy levels $E_t=\{E_{1,t},E_{2,t},\cdots,E_{n,t}\}$, the electricity price P_t and the residential load L_t. Then based on the scheduling strategy, the charging/discharging action $a_t=\{a_{1,t},a_{2,t},\cdots,a_{n,t}\}$ of the battery is decided. The goal of the scheduling strategy is to charge a mass of energy if the electricity price is low and discharge at a high price to satisfy the load requirements and reduce the cumulative charging cost for consumers.

2.1 Battery model

The battery energy model is expressed as follows [28,34]:

$$E_{i,t+1}=E_{i,t}-\eta_i(a_{i,t})a_{i,t} \tag{1}$$

where i represents the battery index. If $a_{i,t}<0$, the battery is charging, and if $a_{i,t}>0$, it is discharging. $a_{i,t}=0$ represents that the battery is idle. We assume that $a_{i,t}$ (kW) is calculated hourly and hence equal to the value of battery energy (kWh). $\eta_i(a_{i,t})$ denotes the efficiency of charging actions, which depend on the loss of auxiliary equipment in the battery system such as transformers, inverters, and transmission lines. It can be defined as

$$\eta_i(a_{i,t})\begin{cases}1, & a_{i,t}\geqslant 0\\ \eta_{i,0}-\xi_i\left|a_{i,t}\right|/a_{i,\text{rate}}, & a_{i,t}<0\end{cases} \tag{2}$$

where $\eta_{i,0}$ and ξ_i are charging parameters of the battery i. $a_{i,\text{rate}}$ is the rated power output.

In order to prolong the lifetime and guarantee the safety of the battery, the energy stored in the battery must be constrained in a safe zone as

$$E_{i,\min}\leqslant E_{i,t}\leqslant E_{i,\max}. \tag{3}$$

Besides, the hourly charging/discharging power is also limited to avoid damage as

$$a_{i,\min}\leqslant a_{i,t}\leqslant a_{i,\max}. \tag{4}$$

2.2 MDP formulation

MDP offers a basic framework for action deciding in uncertain environments where the historic states have no relation to the current states. An MDP can be described as a five-tuple $\langle S,A,P,R,\gamma\rangle$, where S and A denote the entire state and action space, respectively, P represents the state transition probability between states, $R:S\times A\to\mathbf{R}$ is the reward function which maps state-action pairs to rewards, and $\gamma\in[0,1]$ is the discount factor which denotes whether the immediate or future rewards are preferred. We formulate the battery management problem from the consumers' perspective as an MDP with unknown transition probability.

(i) State

At time step t, the system state is $s_t=(E_{1,t},E_{2,t},\cdots,E_{n,t},P_k,L_k,t)$, which can be divided into three types of information: i) The battery energy levels $(E_{1,t},E_{2,t},\cdots,E_{n,t})$observed from the MBESS. ii) The electricity price P_k and the residential load demand L_k. Although they have dynamic fluctuations, their profiles exhibit a degree of periodicity with the period $T=24$ h and typical profiles in a day are given in Fig. 1. In this paper, we assume the electricity price and the load demand have already been predicted (e.g., [35] and [36]). iii) The time index t, used to obtain the price and the load demand.

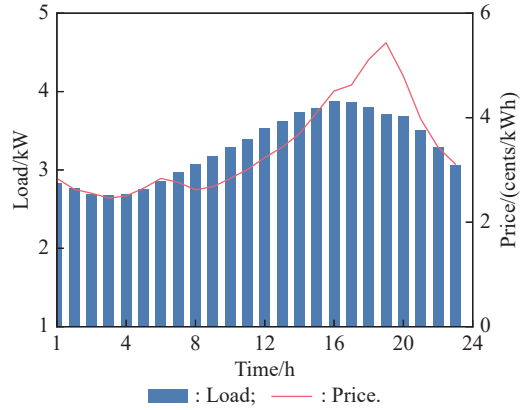

Fig. 1 Typical electricity price and residential load profiles over 24 h

(ii) Action

We assume that the charging/discharging action $a_t=(a_{1,t},a_{2,t},\cdots,a_{n,t})$ at time step t applies continuous charging power and the input constraint (4) must be satisfied for each battery.

(iii) State transition probability

The state transition probability $P(s_{t+1}|s_t,a_t)$ is controlled by action a_t as in (1). Moreover, the electricity price and the load demand are influenced by time index t which makes them time-varying. To simulate the real-world scenario, $P(s_{t+1}|s_t,a_t)$ is considered to be unknown and we propose a model-free algorithm to learn it in Section 3.

(iv) Reward

The immediate reward at time step t is calculated as

$$r(s_t,a_t)=r_t=\begin{cases}-m_1(L_t-a_t)^2+m_2a_tP_t, & E_{\min}\leqslant E_{i,t}\leqslant E_{\max}\\ r_0, & \text{otherwise}\end{cases} \tag{5}$$

where m_1 and m_2 are given weighted coefficients. The first item $(L_t-a_t)^2$ aims to minimize the power purchased from the grid. The second item a_tP_t represents the charging cost from the grid [22,33]. When the battery is charging, this item is negative. When the battery releases energy to residential load devices, this item is positive,

which means the battery system saves an equal cost for the consumer. The penalty item r_0 is a negative constant selected small enough to prevent overcharging and oversupply of the batteries.

(v) Action-value function

The learning objective is to obtain a policy π which decides the charging/discharging action a_t at time step t and maximizes the following performance function:

$$J(\pi)=\mathrm{E}\left[\sum_{t=0}^{T_0}\gamma^t r_{t+1}|s_0,\pi\right] \tag{6}$$

where T_0 is the time step when the MDP terminates. The action-value function or so-called Q-function maps state-action pairs to the cumulative rewards defined as

$$Q^{\pi}(s,a)=\mathrm{E}_{\pi}\left[\sum_{t'=0}^{T_0}\gamma^{t'} r_{t+t'}|s_t=s,a_t=a\right] \tag{7}$$

which changes over time with system state. Then the optimal Q-function is

$$Q^*(s,a)=\max_{\pi}Q^{\pi}(s,a). \tag{8}$$

3. Proposed approach

In this section, we first replace the time-dependent action-value function with time-independent periodic functions and introduce the periodic value iteration. Then, the PDPG algorithm is proposed for the MBESS scheduling problem.

3.1 Periodic action-value function

The optimal action-value function is derived based on the Bellman optimality equation [37] which is given by

$$Q^*(s_t,a_t)=r_t+\gamma\max_{a_{t+1}}Q^*(s_{t+1},a_{t+1}). \tag{9}$$

RL algorithms follow the recursive relationship to estimate the optimal action-value function as follows:

$$Q^{i+1}(s_t,a_t)=r_t+\gamma\max_{a_{t+1}}Q^i(s_{t+1},a_{t+1}). \tag{10}$$

For the formulation of the MDP, we can see that the state s_t is time-variant so that the optimal action-value function $Q^*(s,a)$ consequently varies over time. For different time index t, the optimal action-value function is different despite the same battery energy, electricity price and residential load. This makes it challenging and complex to obtain $Q^*(s,a)$ and the optimal MBESS management becomes a time-varying optimization problem. To tackle this difficulty, we define the new time-invariant state $\bar{s}_t=(E_{1,t},E_{2,t},\cdots,E_{n,t},P_k,L_k)$. In the meantime, the original time-varying optimal action-value function is replaced by a series of time-independent periodic functions $Q_k^*\left(\bar{s}_t,a_t\right)=Q^*(s_t,a_t)(k=\mathrm{mod}\,(t,T)=0,1,\cdots,T-1)$. Then the Bellman optimality function (9) is rewritten as

$$\begin{cases}Q_{T-1}^*\left(\bar{s}_t,a_t\right)=r_t+\gamma\max\limits_{a_{t+1}}Q_0^*\left(\bar{s}_{t+1},a_{t+1}\right)\\ Q_k^*\left(\bar{s}_t,a_t\right)=r_t+\gamma\max\limits_{a_{t+1}}Q_{k+1}^*\left(\bar{s}_{t+1},a_{t+1}\right),\\ \quad k=T-2,T-3,\cdots,0.\end{cases} \tag{11}$$

Therefore, the optimal charging/discharging policy is determined by greedy strategies as

$$a^*=\max_{a\in A}Q_k^*\left(\bar{s},a\right). \tag{12}$$

The iterative periodic equations are generalized as

$$\begin{cases}Q_{T-1}^{i+1}\left(\bar{s}_t,a_t\right)=r_t+\gamma\max\limits_{a_{t+1}}Q_0^i\left(\bar{s}_{t+1},a_{t+1}\right)\\ Q_k^{i+1}\left(\bar{s}_t,a_t\right)=r_t+\gamma\max\limits_{a_{t+1}}Q_{k+1}^i\left(\bar{s}_{t+1},a_{t+1}\right),\\ \quad k=T-2,T-3,\cdots,0.\end{cases} \tag{13}$$

In this paper, we assume $\gamma<1$, that is to say, we pay more attention to the immediate reward than the future reward.

Then the convergence analysis of the above periodic value iterations is given as follows.

Theorem 1 Given an initial Q-function Q_0^0, for $i=0,1,\cdots,$and $k=0,1,\cdots,T-1$, let Q_k^{i+1} be obtained by (13). Then iterative periodic Q-functions Q_k^i converge to their optimums, i.e.,

$$\lim_{i\to\infty}Q_k^{i+1}\left(\bar{s},a\right)=Q_k^*\left(\bar{s},a\right). \tag{14}$$

Proof Consider the case $k=0$. Expanding the right side of recursive equations (13), then we have

$$Q_0^{i+1}\left(\bar{s}_t,a_t\right)=$$
$$\max_{a_t\in A}\cdots\max_{a_{t+T-1}\in A}\left[\sum_{t'=0}^{T-1}\gamma^{t'}r\left(\bar{s}_{t+t'},a_{t+t'}\right)+\gamma^T Q_0^i\left(\bar{s}_{t+T},a_{t+T}\right)\right]. \tag{15}$$

We also expand the Bellman optimality equation (11) based on the same principle as

$$Q_0^*\left(\bar{s}_t,a_t\right)=$$
$$\max_{a_t\in A}\cdots\max_{a_{t+T-1}\in A}\left[\sum_{t'=0}^{T-1}\gamma^{t'}r\left(\bar{s}_{t+t'},a_{t+t'}\right)+\gamma^T Q_0^*\left(\bar{s}_{t+T},a_{t+T}\right)\right]. \tag{16}$$

Then subtracting Q_0^* from Q_0^{i+1} we have

$$\left|Q_0^{i+1}\left(\bar{s}_t,a_t\right)-Q_0^*\left(\bar{s}_t,a_t\right)\right|\leqslant$$
$$\max_{a_t\in A}\cdots\max_{a_{t+T-1}\in A}\gamma^T\left|Q_0^i\left(\bar{s}_{t+T},a_{t+T}\right)-Q_0^*\left(\bar{s}_{t+T},a_{t+T}\right)\right|\leqslant$$
$$\gamma^T\left\|Q_0^i\left(\bar{s}_{t+T},a_{t+T}\right)-Q_0^*\left(\bar{s}_{t+T},a_{t+T}\right)\right\|_{\infty}\leqslant\cdots\leqslant$$
$$\gamma^{(i+1)T}\left\|Q_0^0\left(\bar{s}_{t+T},a_{t+T}\right)-Q_0^*\left(\bar{s}_{t+T},a_{t+T}\right)\right\|_{\infty}. \tag{17}$$

Since Q_0^0 and Q_0^* are obviously finite and $\gamma<1$, when $i\to\infty$, $Q_0^{i+1}\left(\bar{s}_t,a_t\right)\to Q_0^*\left(\bar{s}_t,a_t\right)$. Combining (11) and (13), we can derive that when $i\to\infty$, $Q_k^{i+1}\left(\bar{s}_t,a_t\right)\to Q_k^*\left(\bar{s}_t,a_t\right),k=1,2,\cdots,T-1$. Then the convergence proof of the periodic value iteration is completed. □

In this paper, we take periodic electricity prices and residential load into consideration. Therefore, the system dynamics model is time-dependent. In this case, it is intractable to obtain the optimal action-value function, and the iteration process may not converge. Given that the system model exhibits a degree of periodicity, we remove the time index from the system state and utilize a series of time-invariant periodic action-value functions to describe every period of a day. The original complex problem is transformed into a time-independent optimization problem. The periodic action-value functions converge to their optimums through an iterative way, and hence the optimal charging/discharging strategy can be derived. In [33] and [34], the concept of the periodic value function has been utilized in energy storage systems. However, in [34], the partition of the state space and the determination of membership parameters rely on expert experience, which has subjectivity and uncertainty. In [33], the single-battery scheduling optimization was proposed, and multi-battery management has not been considered.

3.2 PDPG algorithm

The DPG algorithm uses deterministic policy gradient with continuous actions, which outperforms more efficiently than usual stochastic policies. Our PDPG algorithm combines the periodic action-value function and the DPG algorithm with off-policy and actor-critic architecture. With the continuous action space, the policy is moved toward the gradient of the action-value function rather than maximizing it. The proposed algorithm contains multiple critics for different periods and offers scheduling policies for each period. Each critic or actor function is represented by separate neural networks. These networks are connected in chronological order.

The actor functions $A_k\left(\bar{s};\mu_k\right)$ ($k=0,1,\cdots,T-1$) take the system state as input and output deterministic policies, parameterized by μ_k. The critic functions $Q_k\left(\bar{s},a;\theta_k\right)$ take the state and selected actions as inputs to approximate action-value functions, parameterized by θ_k. The actor networks are optimized by updating the policy parameters toward the performance gradient as

$$\nabla_{\mu_k}J_k(A_k)=\mathrm{E}_{\bar{s}\sim\rho^{A_k}}\left[\nabla_{\mu_k}A_k\left(\bar{s};\mu_k\right)\nabla_a Q^{A_k}\left(\bar{s},a\right)\big|_{a=A_k\left(\bar{s};\mu_k\right)}\right] \tag{18}$$

where ρ^{A_k} represents the stationary state distribution and the true Q-function $Q^{A_k}\left(\bar{s},a\right)$ is approximated by the critic function $Q_k\left(\bar{s},a;\theta_k\right)$. The critic networks uses temporal-difference learning to iteratively update parameters, and after the iteration converges, $Q_k\left(\bar{s},a;\theta_k\right)\approx Q_k^*\left(\bar{s},a\right)$. The target action-value y_t at time step t is defined based on $Q_{k+1}\left(\bar{s},A_k\left(\bar{s}_t;\mu_k\right);\theta_{k+1}\right)$ as

$$y_t=\begin{cases} r_t+\gamma Q_0(t+1), & k=T-1\\ r_t+\gamma Q_{k+1}(t+1), & \text{otherwise}\end{cases} \tag{19}$$

where $Q_k(t)=Q_k\left(\bar{s}_t,A_k\left(\bar{s}_t;\mu_k\right);\theta_k\right)$. Based on the Bellman equation, the temporal-difference error (TD-error) δ_t is the difference between two sides without the expected value which is defined as

$$\delta_t=y_t-Q_k\left(\bar{s}_t,a_t;\theta_k\right). \tag{20}$$

Then the critic network parameter θ_k is updated according to the gradient descent method to minimize the TD-error as

$$\theta_k\leftarrow\theta_k+\alpha_\theta\delta_t\nabla_{\theta_k}Q_k\left(\bar{s}_t,a_t;\theta_k\right) \tag{21}$$

where α_θ is the learning rate.

Note that the target value depends on the parameters of critic networks and is updated during each iteration. In this case, the training process may be unstable and even divergent. To alleviate adverse impacts caused by data limitation and stabilize the training process, as suggested in [13] and [15], the target networks with the soft update mechanism for generating the target value y_t are introduced. We use neural networks with parameters $\widehat{\theta}_k$ and $\widehat{\mu}_k$ to approximate T target critic networks and actor networks which slowly synchronize to θ and μ during every training iteration as

$$\widehat{\theta}_k\leftarrow\tau\theta_k+(1-\tau)\widehat{\theta}_k, \tag{22}$$

$$\widehat{\mu}_k\leftarrow\tau\mu_k+(1-\tau)\widehat{\mu}_k, \tag{23}$$

where $0<\tau\leqslant 1$ is the soft update rate. Then the target action-value (19) is rewritten as

$$y_t=\begin{cases} r_t+\gamma\widehat{Q}_0(t+1), & k=T-1\\ r_t+\gamma\widehat{Q}_{k+1}(t+1), & \text{otherwise}\end{cases} \tag{24}$$

where $\widehat{Q}_k(t)=\widehat{Q}_k\left(\bar{s}_t,\widehat{A}_k\left(\bar{s}_t;\widehat{\mu}_k\right);\widehat{\theta}_k\right)$.

To further improve the stability of the updating process, the experience replay technique is also adopted [38]. We introduce T data sets $D=\{D_0,D_1,\cdots,D_{T-1}\}$ and store the agent's transition experience $\{\bar{s}_t,a_t,\bar{s}_{t+1},r_t\}$ in the corresponding M-sized replay buffer D_k at time step t. The action a_t is chosen based on the ϵ-greedy policy. In every

training step, a K -sized mini-batch of transitions $\left\{\left(\bar{s}_j, a_j, \bar{s}_{j+1}, r_j\right)\right\}_{j=1}^{K}$ is randomly sampled from D_k to calculate the following loss function:

$$L_k(\theta_k) = \frac{1}{K}\sum_{j=1}^{K}\left[\bar{y}_j - Q_k\left(\bar{s}_j, a_j; \theta_k\right)\right]^2. \tag{25}$$

Then the critic parameter θ_k is updated by gradient descent of the loss function as

$$\theta_k \leftarrow \theta_k + \alpha_\theta \nabla_{\theta_k} L_k(\theta_k). \tag{26}$$

This technique has several advantages. Firstly, each transition experience can be potentially utilized multiple times in the updating process and thereby the data efficiency is improved. Secondly, the time correlations among samples are removed which further improves the stability. Thirdly, the behavior distribution is averaged over many historical states, so the updating process is smoothed out.

The actor parameters μ_k are updated toward the gradient calculated with regard to the same mini-batch of transitions:

$$\nabla_{\mu_k} J_k(\mu_k) = \frac{1}{K}\sum_{j=1}^{K}\nabla_{\mu_k} A_k\left(\bar{s}_j; \mu_k\right)\nabla_a Q_k\left(\bar{s}_j, a_j; \theta_k\right)\big|_{a_j = A_k\left(\bar{s}_j; \mu_k\right)}, \tag{27}$$

$$\mu_k \leftarrow \mu_k + \alpha_\mu \nabla_{\mu_k} J_k(\mu_k), \tag{28}$$

where α_μ is the learning rate.

The multiple critic and actor networks are updated iteratively by calculating (25) and (27). We develop a two-stage learning protocol composed of the offline training stage and the online real-time scheduling stage. The offline training stage with an episodic style is presented by Algorithm 1. Fig. 2 shows the schematic diagram of developed offline algorithm architecture. The online networks have the same structures as the offline networks. After Algorithm 1 converges, we load trained parameters θ to online networks. The initial state can be arbitrarily decided within the safe zone. At each step t, the system input $\bar{s}_t$ is composed of the battery energy level, the electricity price, and the residential load. Then, the charging/discharging action is selected as $a_t = A_k\left(\bar{s}_t; \mu_k\right)$.

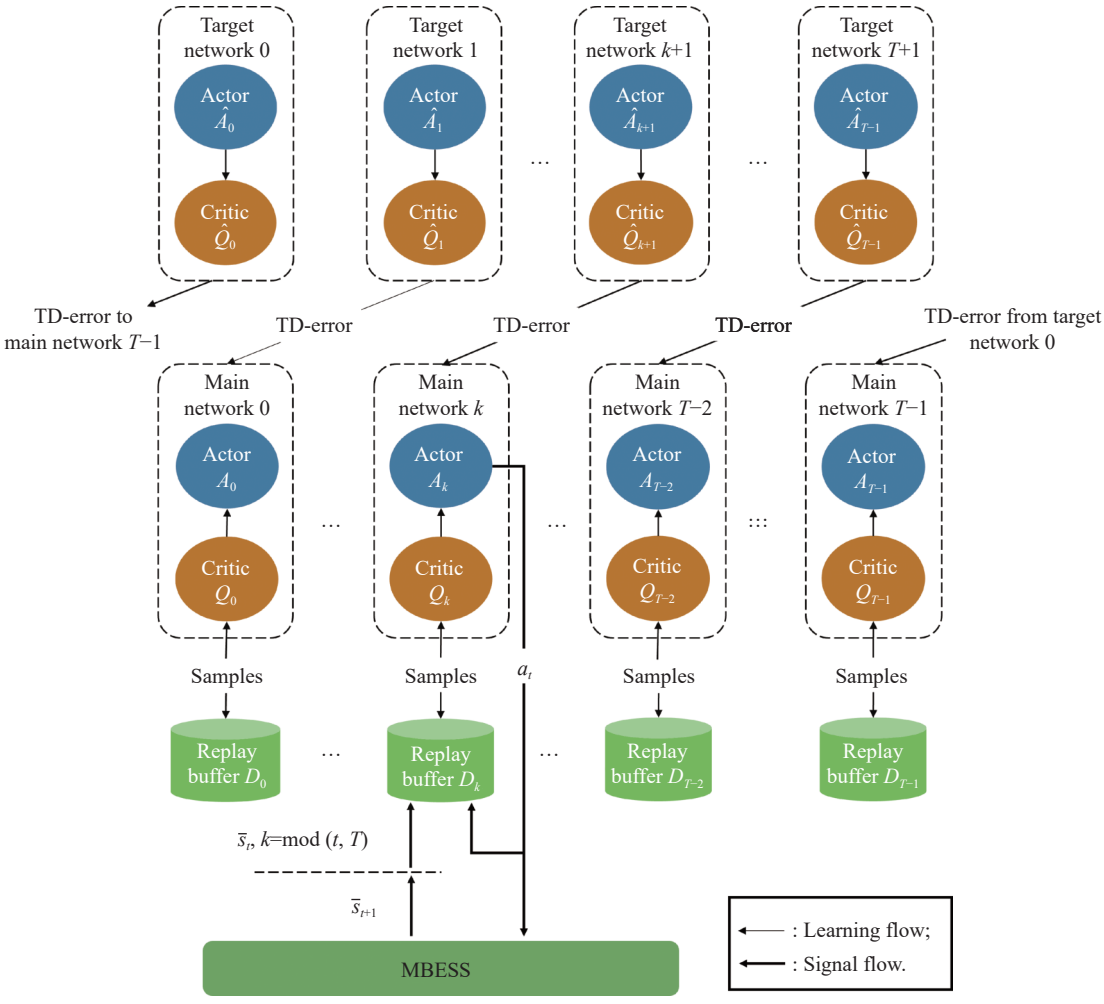

Fig. 2 Schematic diagram of the proposed PDPG architecture

Note that the above updating process is model-free, which does not need the system model because the transition experience can be directly sampled from the emulator and real-world data. It is also off-policy. The behavior policy is selected based on the ε-greedy method that ensures adequate exploration of the state space. The target policy is evaluated based on the greedy method and independent of the behavior policy.

4. Numerical results

In this section, the implementation details of the PDPG algorithm are elaborated for the MBESS charging/discharging problem. We evaluate the proposed approach by numerical case studies and demonstrate its superiority via comparisons.

4.1 Case 1: fixed periodic data

In this case study, we build the simulated MBESS based on the dynamics given in Table 1. The system is composed of four batteries with different charging efficiency parameters and bounds of the battery capacity. Our proposed data-driven approach does not depend on any information of battery parameters and hence could apply to other modeling mechanisms. The electricity prices and the residential load are fixed periodic profiles given in Fig. 1. The charger provides continuous power output for battery charging and discharging. The negative outputs represent the charging process, and the positive values refer to the discharging process.

Table 1 Parameters of the energy storage system

Parameter	Battery 1	Battery 2	Battery 3	Battery 4
η_0	0.958	0.898	0.858	0.798
ξ	0.073	0.073	0.073	0.073
$E_{\min}$/kWh	1.8	1.6	1.0	0.3
$E_{\max}$/kWh	11	9	7	5
$a_{\min}$/kW	−0.9	−0.8	−0.7	−0.6
$a_{\min}$/kW	0.9	0.8	0.7	0.6

In the offline training process, we use three-layer neural networks with random initial parameters to approximate the main and target networks. The training parameters are shown in Table 2. The hidden layer is composed of 64 nodes and fully connected with the input layer. The output layer is fully connected with the hiddenlayer. The training process proceeds with the Adam optimizer. During the first 1 000 training steps, the charging/discharging actions are randomly selected from the action space. After that, the action is selected based on the ε-greedy policy. In the training process, ϵ is reduced from 0.9 to 0.01 during the first 20 000 training steps and remains 0.01 afterward. The average accumulated rewards and the standard deviation over 20 000 episodes learned by PDPG and the classic DPG algorithm are shown in Fig. 3 where the red lines represent the electricity price and the gray stems indicate the residential load. The blue and orange stems indicate the charging/discharging action. The results are based on ten independent trials, and the weights of neural networks are randomly initialized for each trial. The system state of DPG is defined as $s_t = (E_{1,t}, E_{2,t}, \cdots, E_{n,t}, P_k, L_k, t)$. It can be seen that the average reward of PDPG begins to increase gradually from 5 000 episodes. Then, it converges around 45 with slight oscillations after 15 000 episodes. The results show that the developed algorithm succeeds in learning a strategy to achieve high accumulative rewards. Under the same initialization condition and structure of neural networks, the proposed PDPG algorithm converges more quickly and stably, but the convergence process of the DPG algorithm is prone to diverge.

Table 2 Offline training parameters

Parameter	Value
Discount factor γ	0.85
Learning rate α	0.001
Soft update rate τ	0.01
Replay buffer size M	100 000
Minibatch size K	32
MaxStep	168
Weighted coefficients m_1, m_2	0.2, −0.4
Penalty item r_0	−200

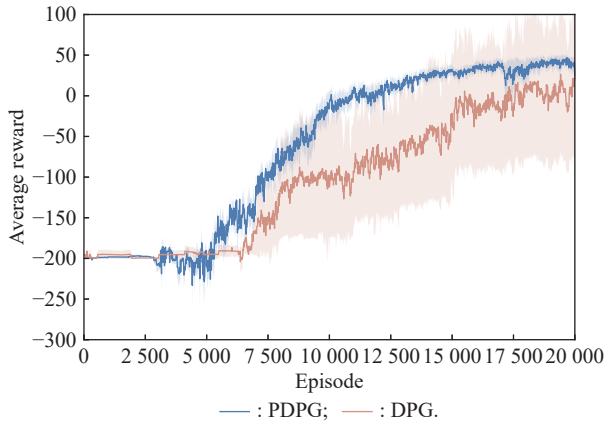

Fig. 3 Average accumulated rewards during the training process

Algorithm 1 Offline training of PDPG

1: for k =0:T−1 **do**
2: Initialize replay buffer D_k with size M
3: Randomly initialize critic function Q_k with weights θ_k
4: Randomly initialize actor function A_k with weights μ_k
5: Initialize target critic function $\widehat{Q}_k$ with weights $\widehat{\theta}_k$=θ_k
6: Initialize target actor function $\widehat{A}_k$ with weights $\widehat{\mu}_k = \mu_k$
7: **end for**
8: for Episode=1:MaxEpisode **do**
9: Obtain the initial state $\bar{s}_0$
10: **for** Time step t=0:MaxStep **do**
11: Obtain the time index $k = \mathrm{mod}(t, T)$
12: With probability ϵ randomly select action a_t otherwise select $a_t = A_k\left(\bar{s}_t; \mu_k\right)$
13: Execute action a_t, collect reward r_t, and move to the next state $\bar{s}_{t+1}$
14: Store transition $\left(\bar{s}_t, a_t, \bar{s}_{t+1}, r_t\right)$ in D_k
15: Randomly sample a mini-batch with size K of transitions $\left\{\left(\bar{s}_j, a_j, \bar{s}_{j+1}, r_j\right)\right\}_{j=1}^{K}$ from D_k
16: Update critic network by (25) and (26)
17: Update actor network by (27) and (28)
18: Update target networks by (22) and (23)
19: **end for**
20: end for

After the offline training, the converged algorithm can be used in real-time charging/discharging scheduling as shown in Fig. 4. The battery energy levels start from $[2.7, 2.4, 2.1, 0.9]$ kWh. In every time step, the total charging/discharging action is calculated as $\sum_{i=1}^{4} a_{i,t}$ and each day shows similar patterns. When electricity price is low, the battery system is charging, and the residential load is satisfied by purchasing more power from utility companies. During peak hours, the battery system tends to discharge, and hence the amount of power bought from utility companies is reduced as much as possible. The gray stems indicate the residential load demand, and one can observe that each battery manages not to over-discharge when the demand is on-peak. In comparison, the policy learned by DPG does not fully discharge or charge when electricity prices fluctuate, resulting in underutilized battery systems. The battery energy levels are shown in Fig. 5 where the dashed lines indicate the safe zones for each battery energy level. The results indicate that each battery is controlled in its safe zone.

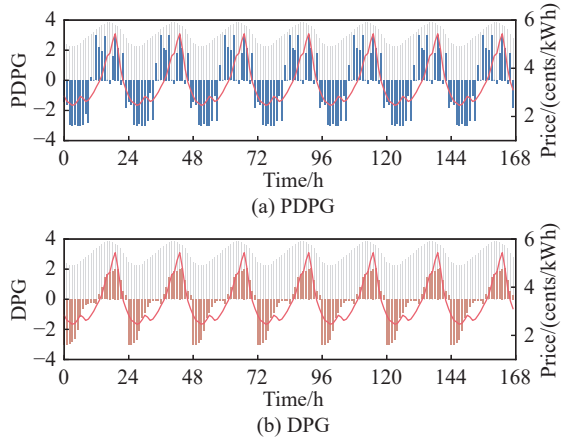

Fig. 4 Comparison of charging/discharging action of the battery system learned by PDPG and DPG over a week

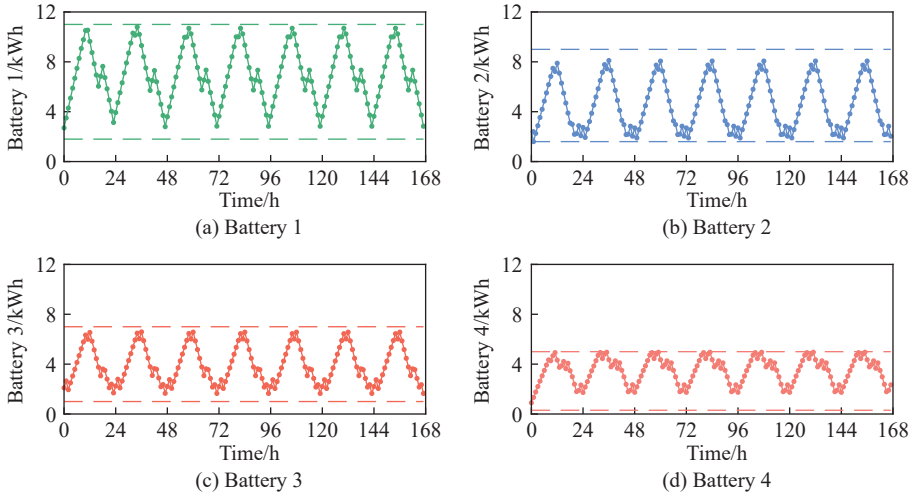

Fig. 5 Battery energy levels over a week

4.2 Case 2: real-world data

In this case study, the real-world electricity price and residential load are considered in simulated energy systems. The real-world data contains 180 days starting from the 1st of March in 2016 collected from California Independent System Operator (CAISO). To avoid overfitting, collected data are divided into the training set and the testing set. Specifically, the first three weeks are used as a training set and the remaining week is selected as a test set for every four consecutive weeks.

To start the training, we first introduce a price predictor model based on prior knowledge. For $\forall t = 0, 1, \cdots$, there exist $\rho = 0, 1, \cdots$, and $k = 0, 1, \cdots, 23$, that satisfies $t = \rho T + k$. Then the predictor is presented by a weighted average predictive filter as $\widehat{P}_t = k_1 P_{k,\rho} + k_2 P_{k,\rho-6}$, where $P_{k,\rho}$ and $P_{k,\rho-6}$ denote actual data in the previous day and the same day last week at time step k, respectively. Let coefficients k_1 and k_2 be 0.838 and 0.156. The load predictor is computed as $\widehat{L}_t = b_1\left(L_{t-1} - \bar{L}_{k-1}\right) + b_2\left(L_{t-2} - \bar{L}_{k-2}\right) + \bar{L}_k$, where L_{t-1} and L_{t-2} represent actual residential load at time step $t-1$ and $t-2$, respectively. $\bar{L}_{k-1}$, $\bar{L}_{k-2}$ and $\bar{L}_k$ are historic averages at time step $k-1$, $k-2$ and k, respectively. Let coefficients b_1 and b_2 be 0.9 and 0.1. The real electricity price and residential load from the 28th of August to the 3rd of September in 2016 are shown in Fig. 6. Both curves show a degree of periodicity and slight fluctuation from day to day. The predicted price and load based on the above models are also depicted. The root mean squared error (RMSE) of the prediction models are 0.475 5 and 0.046 5. Hence, the prediction models are verified practical to the real world. The simulated battery system is also built based on four batteries in case 1.

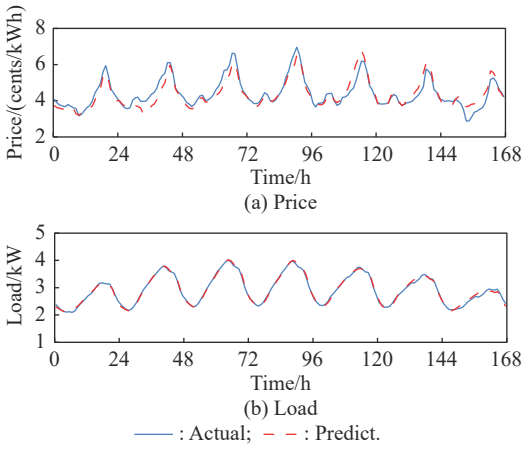

Fig. 6 Real-time data and predictive values over a week

Similar to case 1, we also use three-layer neural networks to approximate the main and target networks. The hidden layer is composed of 64 nodes. The average accumulated rewards and the standard deviation over 30 000 episodes based on ten independent trials are shown in Fig. 7. The convergence process is slower than case 1 because of the complexity of the system dynamics and the consideration of fluctuant price and load demand. After 20 000 episodes, the accumulated rewards converge around 75 and stay steady. Compare with ordinary DPG algorithm, the convergence process of PDPG algorithm is faster and the accumulated rewards are higher. Additionally, PDPG algorithm shows a higher success rate of convergence than DPG algorithm over the ten trials under the same initialization condition and structures of neural networks as in Table 3. Hence, the superiority of the proposed approach is proved.

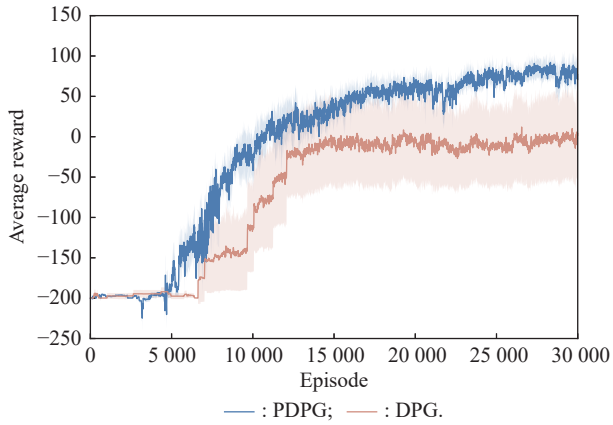

Fig. 7 Average accumulated rewards during the training process

Table 3 Total cost comparison

Evaluation metric	Original	PDPG	DPG
Total cost/cents	2 286.51	**2 085.53**	2 169.63
Saving/%	–	**8.79**	5.12
Success rate of convergence/%	–	**100**	70

The comparison of the total cost over 168 h which is calculated as $\sum_{t=1}^{168}\left(L_t - \sum_{i=1}^{4} a_{i,t}\right)P_t$ is shown in Table 3. The term "original" denotes "no multi-battery system". The proposed PDPG algorithm can save much more costs than the ordinary PDG algorithm. To further illustrate the effectiveness of our approach, the real-time grid power of the battery system over one week which is calculated as $L_t - \sum_{i=1}^{4} a_{i,t}$ is shown in Fig. 8 where the red lines represent the electricity price and the gray stems indicate the residential load. The blue and orange stems are power purchased from grids. The profile of the original con-

sumption is changed. Between each day [1 h, 11 h], the system purchases more electricity than the residential load demand with the additional energy stored in the battery system. Between each day 12 h and the next day 1 h, the purchased energy is significantly decreased because the previously stored power is released. By comparison, the learned grid power by the DPG algorithm does not take full advantage of the difference in electricity prices. The system consumes more electricity than the proposed algorithm when the price is high. Fig. 9 shows the battery energy levels which are all operated in their safe zones. The dashed lines indicate the safe zones for each battery energy level. Similar to the first case, the battery system tends to discharge when the price is high and charge when the price is low.

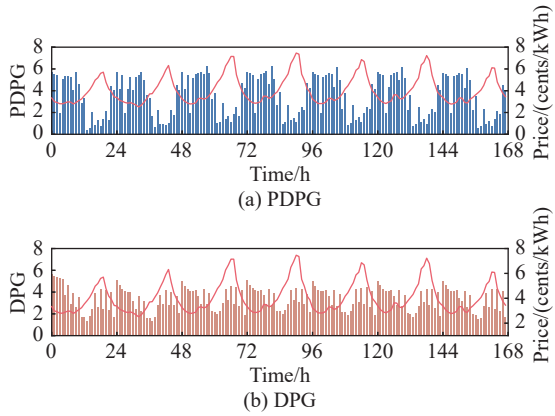

Fig. 8 Comparison of grid power of the battery system learned by PDPG and DPG in real-world

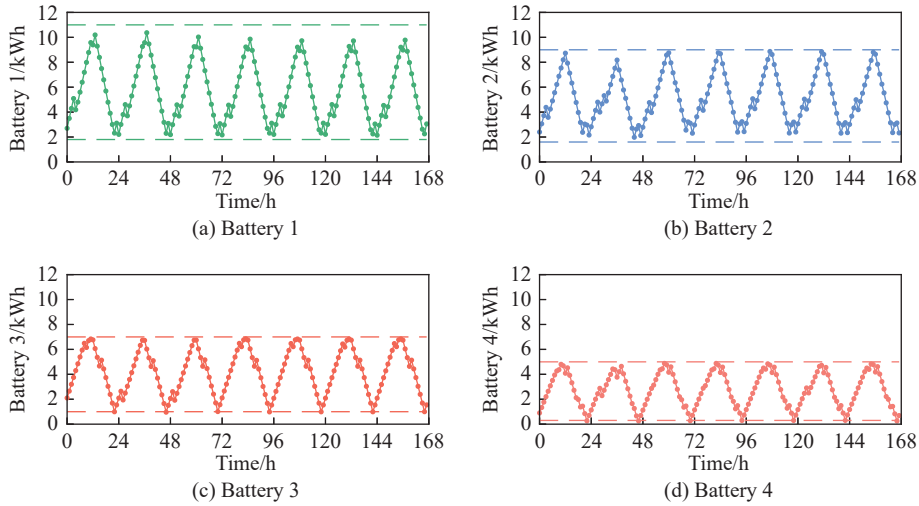

Fig. 9 Battery energy levels over a week

5. Conclusions

In this paper, the MBESS charging/discharging scheduling problem is formulated as an MDP with unknown transition probability. Because both the daily price and load demand exhibit a degree of periodicity, the time-dependent action-value function is replaced by a sequence of periodic time-independent functions. We propose the PDPG algorithm, the DPG with periodic action-value functions, to learn the optimal policy for this problem. A two-stage learning protocol is developed to save consumers' costs in real-time management. The proposed approach has successfully saved costs for consumers as shown in the experimental results of two cases. Additionally, the success rate of convergence of the PDPG algorithm is verified to be higher than the general DPG algorithm.

References

[1] FUSELLI D, DE ANGELIS F, BOARO M, et al. Action dependent heuristic dynamic programming for home energy resource scheduling. International Journal of Electrical Power & Energy Systems, 2013, 48: 148–160.

[2] ERSEGHE T, ZANELLA A, CODEMO C G. Optimal and compact control policies for energy storage units with single and multiple batteries. IEEE Trans. on Smart Grid, 2014, 5(3): 1308–1317.

[3] ALBADI M H, EL-SAADANY E F. A summary of demand response in electricity markets. Electric Power Systems Research, 2008, 78(11): 1989–1996.

[4] SETLHAOLO D, XIA X H. Optimal scheduling of household appliances with a battery storage system and coordination. Energy and Buildings, 2015, 94: 61–70.

[5] LIU C Y, WANG X L, WU X, et al. Economic scheduling model of microgrid considering the lifetime of batteries. IET Generation, Transmission & Distribution, 2017, 11(3): 759–767.

[6] LUNA A C, DIAZ N L, GRAELLS M, et al. Mixed-integer-linear-programming-based energy management system for hybrid PV-wind-battery microgrids: modeling, design, and experimental verification. IEEE Trans. on Power Electronics, 2016, 32(4): 2769−2783.

[7] GAN L K, ZHANG P, LEE J, et al. Data-driven energy management system with Gaussian process forecasting and MPC for interconnected microgrids. IEEE Trans. on Sustainable Energy, 2020, 12(1): 695–704.
[8] ARASTEH F, RIAHY G H. MPC-based approach for online demand side and storage system management in market based wind integrated power systems. International Journal of Electrical Power & Energy Systems, 2019, 106: 124–137.
[9] ZHANG Y, WANG R, ZHANG T, et al. Model predictive control-based operation management for a residential microgrid with considering forecast uncertainties and demand response strategies. IET Generation, Transmission & Distribution, 2016, 10(10): 2367–2378.
[10] HABIB M, LADJICI A A, BOLLIN E, et al. One-day ahead predictive management of building hybrid power system improving energy cost and batteries lifetime. IET Renewable Power Generation, 2019, 13(3): 482–490.
[11] HU K Y, LI W J, WANG L D, et al. Energy management for multi-microgrid system based on model predictive control. Frontiers of Information Technology & Electronic Engineering, 2018, 19(11): 1340–1351.
[12] LU R Z, HONG S H, YU M M. Demand response for home energy management using reinforcement learning and artificial neural network. IEEE Trans. on Smart Grid, 2019, 10(6): 6629–6639.
[13] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518(7540): 529–533.
[14] SILVER D, LEVER G, HEESS N, et al. Deterministic policy gradient algorithms. Proc. of the 31st International Conference on International Conference on Machine Learning, 2014, 32: 387–395.
[15] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning. https://arxiv.org/abs/1509.02971v2.
[16] SCHULMAN J, LEVINE S, ABBEEL P, et al. Trust region policy optimization. Proc. of the 31st International Conference on Machine Learning, 2015. DOI: 10.48550/arXiv.1502.05477.
[17] DONG L, TANG Y F, HE H B, et al. An event-triggered approach for load frequency control with supplementary ADP. IEEE Trans. on Power Systems, 2016, 32(1): 581–589.
[18] DONG L, ZHONG X N, SUN C Y, et al. Adaptive event-triggered control based on heuristic dynamic programming for nonlinear discrete-time systems. IEEE Trans. on Neural Networks and Learning Systems, 2016, 28(7): 1594–1605.
[19] WU Z Q, WEI J, ZHANG F, et al. MDLB: a metadata dynamic load balancing mechanism based on reinforcement learning. Frontiers of Information Technology & Electronic Engineering, 2020, 21(7): 1034–1046.
[20] XU X, JIA Y W, XU Y, et al. A multi-agent reinforcement learning-based data-driven method for home energy management. IEEE Trans. on Smart Grid, 2020, 11(4): 3201–3211.
[21] BAHRAMI S, CHEN Y C, WONG V W. Deep reinforcement learning for demand response in distribution networks. IEEE Trans. on Smart Grid, 2020, 12(2): 1496–1506.
[22] WAN Z Q, LI H P, HE H B, et al. Model-free real-time EV charging scheduling based on deep reinforcement learning. IEEE Trans. on Smart Grid, 2018, 10(5): 5246–5257.
[23] CAO J, HARROLD D, FAN Z, et al. Deep reinforcement learning-based energy storage arbitrage with accurate lithium-ion battery degradation model. IEEE Trans. on Smart Grid, 2020, 11(5): 4513–4521.
[24] YU L, XIE W W, XIE D, et al. Deep reinforcement learning for smart home energy management. IEEE Internet of Things Journal, 2019, 7(4): 2751–2762.
[25] MOCANU E, MOCANU D C, NGUYEN P H, et al. On-line building energy optimization using deep reinforcement learning. IEEE Trans. on Smart Grid, 2018, 10(4): 3698–3708.
[26] GOROSTIZA F S, GONZALEZ-LONGATT F M. Deep reinforcement learning-based controller for SOC management of multi-electrical energy storage system. IEEE Trans. on Smart Grid, 2020, 11(6): 5039–5050.
[27] ZHU F Q, YANG Z P, LIN F, et al. Decentralized cooperative control of multiple energy storage systems in urban railway based on multiagent deep reinforcement learning. IEEE Trans. on Power Electronics, 2020, 35(9): 9368–9379.
[28] HUANG T, LIU D R. A self-learning scheme for residential energy system control and management. Neural Computing and Applications, 2013, 22(2): 259–269.
[29] MBUWIR B V, RUELENS F, SPIESSENS F, et al. Battery energy management in a microgrid using batch reinforcement learning. Energies, 2017, 10(11): 1846.
[30] KIM S, LIM H. Reinforcement learning based energy management algorithm for smart energy buildings. Energies, 2018, 11(8): 2010.
[31] LIU L T, GAURAV S. A solution to time-varying Markov decision processes. IEEE Robotics and Automation Letters, 2018, 3(3): 1631–1638.
[32] VAZQUEZ-CANTELI J R, NAGY Z. Reinforcement learning for demand response: a review of algorithms and modeling techniques. Applied Energy, 2019, 235: 1072–1089.
[33] WEI Q L, LIU D R, SHI G. A novel dual iterative Q-learning method for optimal battery management in smart residential environments. IEEE Trans. on Industrial Electronics, 2014, 62(4): 2509–2518.
[34] ZHU Y H, ZHAO D B, LI X J, et al. Control-limited adaptive dynamic programming for multi-battery energy storage systems. IEEE Trans. on Smart Grid, 2018, 10(4): 4235–4244.
[35] KONG W C, ZHAO Y D, JIA Y W, et al. Short-term residential load forecasting based on LSTM recurrent neural network. IEEE Trans. on Smart Grid, 2017, 10(1): 841–851.
[36] JONSSON T, PINSON P, NIELSEN H A, et al. Forecasting electricity spot prices accounting for wind power predictions. IEEE Trans. on Sustainable Energy, 2012, 4(1): 210–218.
[37] BELLMAN R. Dynamic programming. Science, 1996, 153(3731): 34–37.
[38] LIN L J. Reinforcement learning for robots using neural networks. Pittsburgh: Carnegie Mellon University, 1992.

Biographies

CHENG Guangran was born in 1996. She received her B.S. degree in automation from Nanjing University of Science and Technology, Nanjing, China, in 2018. She is currently working toward her Ph.D. degree in control science and engineering at the School of Automation, Southeast University, Nanjing, China. Her current research interests include reinforcement learning, multi-objective learning, and robot navigation.
E-mail: chenggr@seu.edu.cn.

DONG Lu was born in 1990. She received her B.S. degree in the School of Physics and Ph.D. degree in the School of Automation from Southeast University, Nanjing, China, in 2012 and 2017, respectively. She is currently an associate professor with the School of Cyber Science and Engineering, Southeast University. Her current research interests include adaptive dynamic programming, event-triggered control, nonlinear system control, and optimization.

E-mail: ldong90@seu.edu.cn

YUAN Xin was born in 1989. He received his B.S. degree in electrical engineering and M.S. degree in vehicle operation engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2012 and 2015, respectively, and Ph.D. degree in control science and engineering from Southeast University, Nanjing, China, in 2021. Currently, he is working as a postdoctoral researcher with the School of Automation, Southeast University, Nanjing, China. He was a joint Ph.D. student with the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI, USA, from 2017 to 2018. His current research interests include reinforcement learning, unmanned aerial vehicle, and optimal control.

E-mail: xinyuan@seu.edu.cn

SUN Changyin was born in 1975. He received his B.S. degree in applied mathematics from the College of Mathematics, Sichuan University, Chengdu, China, in 1996, and M.S. and Ph.D. degrees in electrical engineering from Southeast University, Nanjing, China, in 2001 and 2004, respectively. He is currently a professor with the School of Automation, Southeast University, Nanjing, China. His current research interests include intelligent control, flight control, and optimal theory.

E-mail: cysun@seu.edu.cn