Adaptive Model-Based Reinforcement Learning for Fast Charging Optimization of Lithium-Ion Batteries

Yuhan Hao, Qiugang Lu, Xizhe Wang and Benben Jiang, *Member, IEEE*

***Abstract*— Fast charging problem of lithium-ion batteries with minimum-charging time while limiting battery degradation, has been receiving increasing attention and is a critical challenge to battery community. Difficulties in this optimization lie in that: (i) The parameter space of charging strategies is high dimensional while the budget of the experimental cost is often limited; (ii) The evaluation of charging strategies' performance is expensive, and (iii) the degradation process of battery is strongly nonlinear and multiple degradation mechanisms occur simultaneously leading to difficulties for establishing accurate first-principles models. Current methods to address these difficulties are mainly electrochemical model-based optimization and grid search, which are rarely adaptive to battery degradation and/or are of low sample efficiency. In this work, we propose an adaptive model-based reinforcement learning (RL) approach for fast charging optimization while limiting battery degradation, in which a probabilistic surrogate model of differential Gaussian process (GP) is adopted to adaptively describe the degradation of cells. The effectiveness of the proposed approach is demonstrated on PETLION, a high-performance PET-based battery simulator. The results show that (i) compared with the model-free RL method, the proposed adaptive GP-based RL approach possesses superior charging performance and high sample efficiency, and (ii) the proposed method performs well in the handling of degradation constraints on voltage and temperature for dynamically aging batteries with its adaptability to the variations of environment.**

***Index Terms*— Fast charging optimization, Lithium-ion battery, Reinforcement learning, Machine learning, Gaussian process**

I. Introduction

As lithium-ion batteries become ubiquitous in clean power systems, including electric vehicles and smart grids [1], [2], the fast charging of batteries has gained increasing interest in the battery community [3], [4]. Fast charging can be achieved by excessively large currents, but such charging strategies can cause severe degradation to the battery life [5], [6]. Moreover, the relation between charging policy and the resulting charging performance is unknown and expensive to evaluate, which poses significant challenges for the optimization of battery charging for real-world applications.

The design of fast-charging strategies has been widely studied in the literature. For instance, Klein et al. [7] adopted a nonlinear predictive controller to solve the minimum time charging problem with various constraints on battery states. Suthar et al. [8] proposed an optimal charging strategy to minimize mechanical damage to battery particles based on the capacity fade mechanism. A model predictive control (MPC) method was proposed in [9] to design a fast charging profile with inputs, outputs, and constraints taken into account. Jiang et al. [10] applied a data-driven Bayesian optimization approach to solve a constrained minimum-charging time problem with continuous-varied current charging protocols, which outperforms conventional CC-CV methods. However, these aforementioned methods are rarely adaptive to parameter drifting as battery degrades. To this end, the reinforcement learning (RL) framework is employed to enable the adaptive optimization of charging strategies [11]. Specifically, the RL agent iteratively interacts with the environment (battery) by deploying an action (charging current), receiving a reward, and updating the policy (charging strategy) parameters, to maximize a properly predefined cumulative reward. In this process, the charging strategy can be adaptively updated in the presence of environment parameter changes as battery ages [11].

Tabular methods are a class of typical RL algorithms to solve policy iteration, e.g., Q-learning, but the convergence rate of tabular methods is severely deteriorated as the dimension of state-action space increases, known as the curse of dimensionality. In addition, tabular methods are not applicable to continuous state-action space [12]. As a solution to this problem, approximate dynamics programming is proposed for continuous state-action space, in which the Q-table in tabular methods is replaced by a function estimator, e.g., artificial neural network (ANN) [13]. Using deep neural networks as function approximators for RL has achieved success in various fields such as ocean engineering [14], robot navigation [15], and smart IoT system [16]. Actor-critic algorithm is one of such methods that perform well in non-Markovian stochastic processes [17]. In [11], a model-free actor-critic RL framework is proposed for designing fast charging strategy with safety constraints, in which violations of safety constraints are

This work was supported by the Tsinghua–Toyota Joint Research Fund, the National Natural Science Foundation of China (No. 62273197), and the National Key Research and Development Program of China (No. 2022YFE0197600) (Corresponding author: Benben Jiang).

Y. Hao, X. Wang, and B. Jiang is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mails: haoyh19@mails.tsinghua.edu.cn, wangxz18@mails.tsinghua.edu.cn, bbjiang@tsinghua.edu.cn).

Q. Lu is with the Department of Chemical Engineering, Texas Tech University, Lubbock, TX 79409, USA (e-mail: Jay.Lu@ttu.edu).

included in the reward function. It is reported that this model-free RL method shows good adaptability to parameter drifting of the environment but is low in sample efficiency, requiring hundreds or thousands of interactions with environment for policy learning.

In reinforcement learning, two types of efficiency are important: sample efficiency and computing efficiency [18]. Model-free RL methods train the policy without an explicit surrogate model to represent the environment, which often shows good computing efficiency but low sample efficiency [11]. Model-based RL methods learn a surrogate model of the environment and implement policy iterations simultaneously. They perform better in sample efficiency due to fewer interactions with the environment but worse in computing efficiency due to model optimization and prediction [19], [20]. For scenarios where the cost of interaction with the environment is expensive, e.g., the evaluation of battery charging policy, it is appropriate to adopt a model-based RL method to achieve higher sample efficiency. However, the higher sample efficiency achieved by model-based methods is usually at the cost of larger asymptotic bias [21]. There is no guarantee of optimal solution or even convergence of the learning process for model-based methods when the surrogate model is not accurate enough to represent the real environment. Therefore, an appropriate model and training strategy should be designed carefully for certain application problems to achieve superior performance in both sample efficiency and asymptotic bias. For instance, Pong et al. [21] proposed temporal difference models for RL to provide relatively high sample efficiency on a series of continuous control tasks. A deep kernel learning approach [22] was put forward to learn informed kernels to build Gaussian process (GP) for the model-based RL. In summary, the following three challenges are present for optimizing the battery charging policies: (1) This optimization problem is black-box where the relation between charging strategy and resultant charging performance is unknown and expensive to evaluate; (2) Current methods cannot provide a smart optimal charging policy that is adaptive to battery aging; and (3) Existing RL methods for optimizing the charging policy are low in sample efficiency.

In this work, we study the usage of model-based RL to optimize fast charging strategies for dynamically aging batteries with degradation constraints. To reduce battery degradation, constraints on voltage and temperature are imposed since battery degradation mechanisms, such as SEI growth [23], [24], are often related to the operating range of battery temperature, voltage, and current density [10], [25]. An adaptive model-based RL framework is developed for the optimization of this minimum-charging time problem in continuous action-state space, in which a probabilistic GP is applied to model the cell environment, and a differential GP is employed to model the degradation of cells.

The main contributions of this work are as follows:

(i) An adaptive model-based RL approach is proposed to optimize the minimum-time charging strategies in continuous current space with constraints on battery degradation. Compared with model-free RL methods, the proposed model-based charging method possesses superior performance and high sample efficiency, which can considerably reduce the experimental cost for fast charging design.

(ii) The proposed approach utilizes a probabilistic surrogate model based on differential GP to adaptively represent the dynamic aging of batteries, and shows a superior charging performance in handling the degradation process.

The rest of this article is organized as follows. The RL framework is briefly revisited in Section II. The proposed adaptive model-based RL approach is detailed in Section III, followed by a formulation of battery charging problem in Section IV. The effectiveness of the proposed fast charging approach is demonstrated and discussed in Section V. The conclusion of this work is summarized in Section VI.

II. Reinforcement Learning Fundamentals

The key elements of reinforcement learning including Markov decision process (MDP) and actor-critic framework are briefly reviewed in this section. More detailed information about MDP can be found in [19], [26], [27].

A. Markov Decision Process

MDP solves the problem of finding the best policy that maximizes the total cumulative rewards received from the environment. In MDP, we use a vector $\mathbf{s}_t \in \mathbb{S}$ to describe the state of environment at step $t \in \mathbb{R}^+$, where $\mathbb{S}$ stands for the state space. The action to be taken at step t is denoted by vector $\mathbf{a}_t \in \mathbb{A}$, where $\mathbb{A}$ represents the action space. Basic knowledge of the environment includes state-transition probability $p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ and instant reward $r_{t+1} = r(\mathbf{s}_t, \mathbf{a}_t)$. A policy is a mapping from state to action $\pi : \mathbb{S} \rightarrow \mathbb{A}$, and it decides the action to be taken based on the state. The total discounted cumulative reward is quantitatively defined as

$$G_t = \sum_{k=0}^{+\infty} \gamma^k r_{t+1+k}, \tag{1}$$

where $\gamma \in [0, 1]$ is the discounting factor. The expected total discounted cumulative reward from $\mathbf{s}_t$ onward is regarded as the MDP's objective function, which is shown to be

$$V^\pi(\mathbf{s}_t) = \mathbb{E}\left[G_t|\mathbf{s}_t\right], \tag{2}$$

where $V^\pi(\mathbf{s}_t)$ is the state function, and π is the underlying policy. The optimal state function is often obtained indirectly from the state-action function, or Q-function, defined as

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t) = \mathbb{E}\left[G_t|\mathbf{s}_t, \mathbf{a}_t\right]. \tag{3}$$

Based on the Bellman equation, it follows that

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}). \tag{4}$$

The objective of MDP is to find the optimal policy π to maximize the state-action function or state function, i.e.,

$$Q^*(\mathbf{s}_t, \mathbf{a}_t) = \arg\max_{\pi} Q^\pi(\mathbf{s}_t, \mathbf{a}_t), \tag{5}$$

$$V^*(\mathbf{s}_t) = \arg\max_{\mathbf{a}_t \in \mathbb{A}} Q^*(\mathbf{s}_t, \mathbf{a}_t). \tag{6}$$

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2023.3257299

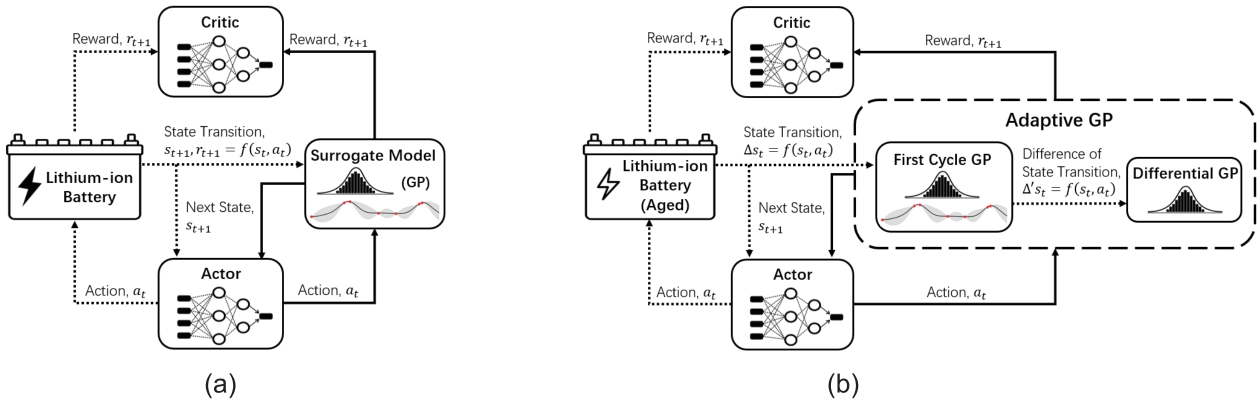


Fig. 1. Model-based reinforcement learning framework for fast-charging: (a) Static model-based RL with no adaptability, and (b) adaptive GP model-based RL for a changing environment. The dotted arrows represent the workflow of a model-free RL method and the solid arrows stands for the workflow of the model-based RL methods. For clarity and simplification, the interaction between actor and critic during calculation of loss function, backward propagation, and parameter update is omitted in this schematic.

B. Actor-Critic Method

The actor-critic method is effective in handling continuous state and action spaces. Two neural networks, defined as actor network $\pi(\mathbf{s}_t|\theta^{\pi})$ and critic network $Q(\mathbf{s}_t, \mathbf{a}_t|\theta^{Q})$, parameterized respectively by θ^{π} and θ^{Q}, are used to represent the actor and critic. The deep deterministic policy gradient is used to learn the network parameters.

1) Critic: The critic network is used to estimate the Q-function based on the policy provided by the actor. The network parameters are updated according to the follows. Consider that at step t, the state is $\mathbf{s}_t$, then an action $\mathbf{a}_t$ with exploration noise is applied to the environment, which returns the instant reward r_{t+1} and next-step state $\mathbf{s}_{t+1}$. Then the loss function for training the critic network is formulated as minimizing the difference between critic network and the temporal difference (TD) target obtained with n random samples from the replay buffer [19], [27]:

$$L(\theta^{Q}) = \frac{1}{N}\sum_{t=1}^{N}(y_t - Q(\mathbf{s}_t, \mathbf{a}_t|\theta^{Q}))^2, \tag{7}$$

where y_t $(t = 1, 2, \ldots, N)$ is the TD target, defined as

$$y_t = r_{t+1} + \gamma Q(\mathbf{s}_{t+1}, \pi(\mathbf{s}_{t+1}|\theta^{\pi})|\theta^{Q}). \tag{8}$$

The critic network parameters are updated as follows

$$\theta^{Q}_{k+1} = \theta^{Q}_{k} - \eta_Q \nabla_{\theta^Q} L(\theta^{Q}), \tag{9}$$

where k stands for the iterates of gradient descent algorithm, and η_Q is the learning rate of the critic network.

2) Actor: The actor network returns the optimal action based on current state. The network parameters are updated to maximize the cumulative expected reward (i.e., policy gradient) as maximize V equals minimize the loss $-V$

$$L(\theta^{\pi}) = -V^{\pi}(\mathbf{s}_t), \tag{10}$$

$$\theta^{\pi}_{k+1} = \theta^{\pi}_{k} - \eta_{\pi} \nabla_{\theta^{\pi}} L(\theta^{\pi}), \tag{11}$$

where η_{π} is the learning rate of the actor network, and the cumulative expected reward $V^{\pi}(\mathbf{s}_t)$ is replaced by its estimate $Q(\mathbf{s}_i, \mathbf{a}_i|\theta^{Q})$ provided by the critic in real implementation.

III. The Proposed Adaptive Model-Based RL For Fast Charging Optimization

In this section, we demonstrate the detailed methodology to establish static and adaptive GP models as the surrogate of a generic RL environment.

A. Probabilistic Surrogate Model Based on GP

As mentioned above, the learning process of a model-free RL requires numerous interactions with the environment, which is not appropriate for scenarios where the cost of interactions with environment is expensive, just as the case of fast-charging optimization for lithium-ion batteries. To reduce the evaluation cost, we first build a probabilistic surrogate model to represent the environment based on the data from only a few interactions with environment, after which the training of RL only needs to interact with the model instead of the environment. This method is termed as model-based RL since we use a surrogate model to represent the environment. Compared to model-free RL, the presence of an environment model can considerably improve the sample efficiency in training RL.

For typical RL, the input to the environment is the decided action $\mathbf{a}_{i-1}$ from the RL agent, and the output of the environment is the next-step state $\mathbf{s}_i$ and reward. Specific to our study, the RL environment is the battery simulator, the action is the designed charging currents. A GP is employed to build the probabilistic surrogate model, where the tuples $\mathbf{x}_i = (\mathbf{s}_{i-1}, \mathbf{a}_{i-1}) \in \mathbb{R}^{D+F}$ and the differences $\Delta_i = (\Delta\mathbf{s}_i, \Delta t_i) \in \mathbb{R}^{D+1}$ are used as training inputs and outputs, respectively. Here, D and F stand for the dimension of state and action space, respectively, t_i is the i-th RL step, and $\Delta\mathbf{s}_i = \mathbf{s}_i - \mathbf{s}_{i-1}$ represents the difference of state between the current and previous steps. Specifically, the GP model is described as

$$f(\mathbf{x}) \sim \mathrm{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \tag{12}$$

where $m(\mathbf{x})$ is the mean that is often specified to be zero, and $k(\mathbf{x}, \mathbf{x}')$ is the covariance function. A radial-basis function (RBF) kernel is used in this work, with its hyperparameters learned from data via L-BFGS [28]. The learned GP model then replaces the environment in the actor-critic framework to predict the state transition information as

$$\mu_f(\Delta_*|\mathbf{x}_*) = \mathbf{k}_*^\top \mathbf{K}^{-1}\mathbf{y} = \mathbf{k}_*^\top \beta, \tag{13}$$

$$\sigma_f^2(\Delta_*|\mathbf{x}_*) = k_{**} - \mathbf{k}_*^\top \mathbf{K}^{-1}\mathbf{k}_*, \tag{14}$$

where $\mathbf{k}_* := k(\mathbf{X}, \mathbf{x}_*)$, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, $k_{**} := k(\mathbf{x}_*, \mathbf{x}_*)$, $\beta := \mathbf{K}^{-1}\mathbf{y}$, $\mathbf{y} = [\Delta_1, \Delta_2, \ldots, \Delta_n]^\top$, and $\mathbf{K}$ is the Gram matrix with entries $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

B. Adaptive Surrogate Model Based on Differential GP for Dynamically Changing Environment

The previous section presents the usage of GP for modeling a stable environment. However, such static GP cannot well model a dynamically changing environment (e.g., battery aging). To this end, in this section, we propose an adaptive surrogate model based on differential GP to capture the changes of the environment.

Likewise, the tuples $\tilde{\mathbf{x}}_i = (\tilde{\mathbf{s}}_{i-1}, \tilde{\mathbf{a}}_{i-1}) \in \mathbb{R}^{D+F}$ and the second-order difference $\Delta'_i = (\Delta'\tilde{\mathbf{s}}_i, \Delta'\tilde{t}_i) \in \mathbb{R}^{D+1}$ are taken as inputs and outputs of the adaptive surrogate model based on differential GP, where $(\Delta'\tilde{\mathbf{s}}_i, \Delta'\tilde{t}_i) = (\Delta\tilde{\mathbf{s}}_i, \Delta\tilde{t}_i) - \mu_f(\Delta_i|\tilde{\mathbf{x}}_i)$, where the superscript “′” denotes the second-order difference and the symbol “~” denotes the dynamically changing environment, and $\mu_f(\Delta_i|\tilde{\mathbf{x}}_i)$ is the expected state transition information predicted by GP for the initial environment. Similarly, the L-BFGS algorithm is performed to estimate the hyperparameters of the differential GP model. After that, the adaptive GP model can be used to predict the dynamically changing environment as

$$\mu_{f'}(\Delta'_*|\tilde{\mathbf{x}}_*) = \tilde{\mathbf{k}}_*^\top \tilde{\mathbf{K}}^{-1}\tilde{\mathbf{y}} = \tilde{\mathbf{k}}_*^\top \tilde{\beta}, \tag{15}$$

$$\sigma_{f'}^2(\Delta'_*|\tilde{\mathbf{x}}_*) = \tilde{k}_{**} - \tilde{\mathbf{k}}_*^\top \tilde{\mathbf{K}}^{-1}\tilde{\mathbf{k}}_*, \tag{16}$$

where $\tilde{\mathbf{k}}_* := k(\tilde{\mathbf{X}}, \tilde{\mathbf{x}}_*)$, $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_n]$, $\tilde{k}_{**} := k(\tilde{\mathbf{x}}_*, \tilde{\mathbf{x}}_*)$, $\tilde{\beta} := \tilde{\mathbf{K}}^{-1}\tilde{\mathbf{y}}$, $\tilde{\mathbf{y}} = [\Delta'_1, \Delta'_2, \ldots, \Delta'_n]^\top$, and $\tilde{\mathbf{K}}$ is the Gram matrix with entries $\tilde{\mathbf{K}}_{ij} = k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$.

Once the differential GP is well trained, we can integrate the GP model built from the initial environment states with the differential GP model to adaptively track the changes of the environment. The predicted state transition by the adaptive GP model is

$$\mu_{\tilde{f}}(\tilde{\Delta}_*|\tilde{\mathbf{x}}_*) = \mu_f(\Delta_*|\tilde{\mathbf{x}}_*) + \mu_{f'}(\Delta'_*|\tilde{\mathbf{x}}_*). \tag{17}$$

In summary, the differential GP is designed to fit the residual of state transition between initial GP prediction and the actual state transition of the new environment, thus can capture the changes in the environment. A detailed workflow of our algorithm is summarized in the following section.

C. The Proposed Adaptive GP-based RL Approach

Fig. 1 (a) shows the workflow of the proposed static GP-based RL framework for an unchanging environment. Specifically, the first N interaction data of the agent with the real battery system are used to train the surrogate model (dotted lines). After that, the RL interacts only with the surrogate model to train the actor-critic networks until convergence. The detailed implementation of this method is provided in Algorithm 1.

Algorithm 1 Static GP model-based RL

```
Randomly initialize critic and actor networks;
for $n = 1$ to $N$ do
    Provide a random initial state $\mathbf{s}_0$ of the environment;
    $i \leftarrow 0$;
    repeat
        Apply $\mathbf{a}_i = \pi(\mathbf{s}_i|\theta^\pi) + \varepsilon_i$ to the environment to
        transit into the next state $\mathbf{s}_{i+1}$ with instant reward
        $r_{i+1}$, where $\varepsilon_i$ stands for exploration noise;
        Update GP based on L-BFGS algorithm;
        Update critic and actor based on (9) and (11);
        $i \leftarrow i + 1$;
    until Reach the final state of the environment
end for
repeat
    Provide a random initial state $\mathbf{s}_0$ of GP;
    $i \leftarrow 0$;
    repeat
        Apply $\mathbf{a}_i = \pi(\mathbf{s}_i|\theta^\pi)$ to GP to transit into the next
        state $\mathbf{s}_{i+1}$ with instant reward $r_{i+1}$;
        Update critic and actor based on (9) and (11);
        $i \leftarrow i + 1$;
    until Reach the final state of GP
until Convergence
```

Fig. 1 (b) illustrates the structure of the proposed adaptive GP-based RL framework for a dynamically changing environment. Specifically, before the beginning of a new cycle, we initialize a new agent with the parameters of the old agent which has been trained in the first cycle. For a new cycle, The first N' episodes are conducted in a manner where the agent interacts with the real battery system, to create some true data for training the differential GP. Once the differential GP is trained, we will combine it with the static GP that was previously trained in the first cycle to predict the next-step state and reward of the new environment. From then on, the RL will perform with the GP surrogate model for the rest episodes of a cycle. The pseudo code of the adaptive GP-based RL is given in Algorithm 2.

IV. Fast Charging Optimization

In this section, we first introduce a porous electrode theory (PET)-based battery simulator that will be used to evaluate the proposed RL-based fast charging methods, followed by the formulation of the battery fast charging problem.

A. Electrochemical Model

We summarize the key elements of the PET model in this subsection. More details about the PET model can be found in [23]–[25], [29]. In a PET model, the diffusion of lithium ions in each solid particle is described by the Fick's second

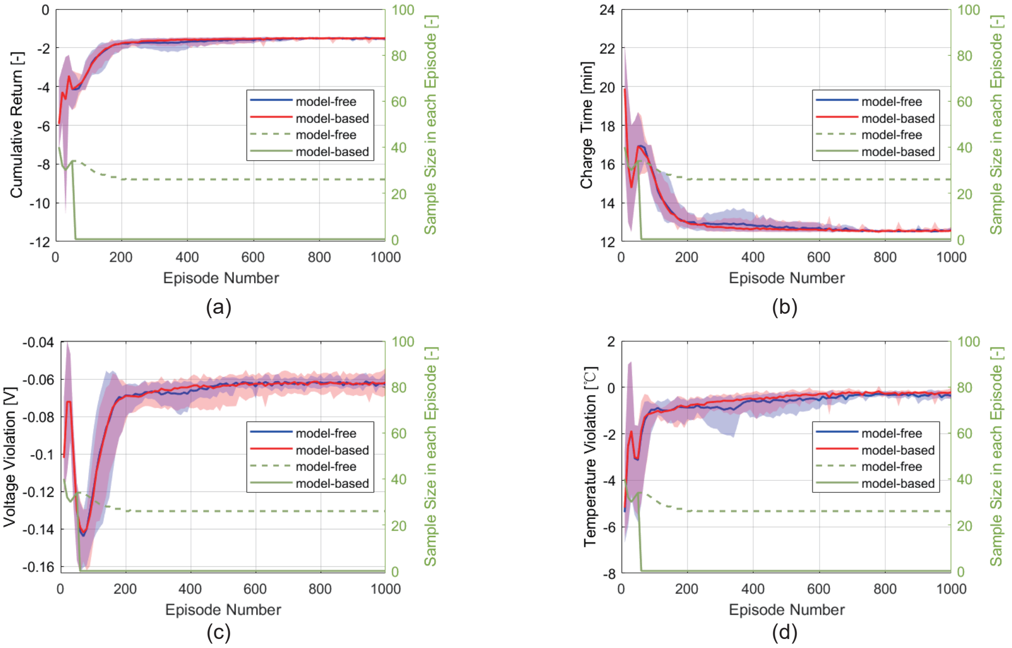


Fig. 2. Performance of the minimum time charging optimization using the proposed GP model-based RL approach and the model-free RL method at $T_{\text{amb}} = 25°$C, where the shaded region stands for the range between the maximum and minimum values of a variable, and the line stands for the mean value: (a) cumulative return, (b) charge time, (c) voltage violation, and (d) temperature violation. The sample size used for training in each episode is represented by the number of state transitions of the simulator referring to the right axis.

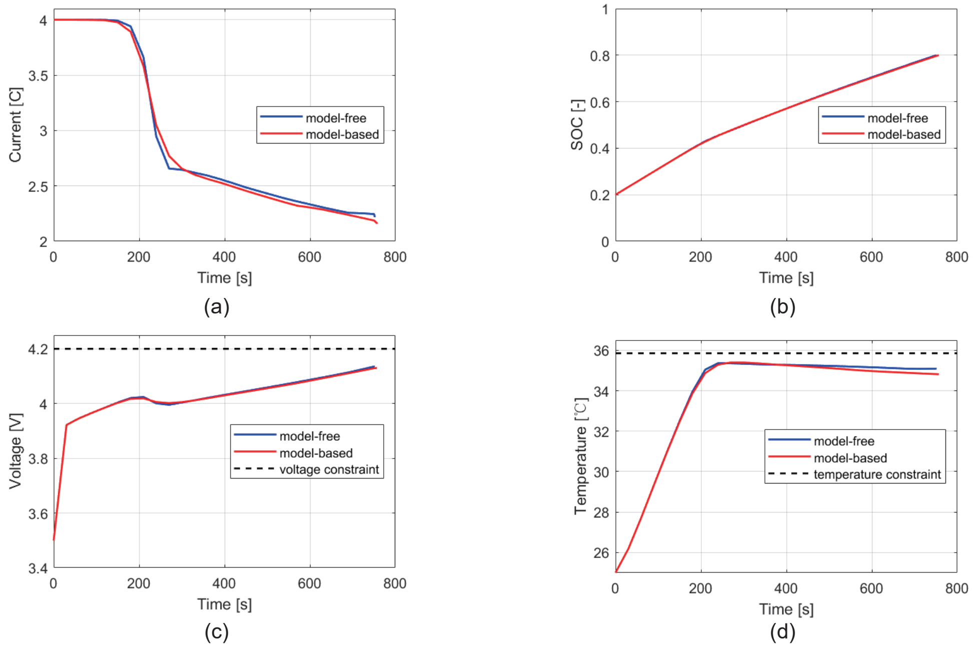


Fig. 3. The optimal charging profiles provided by the model-free RL and the GP model-based RL at $T_{\text{amb}} = 25°$C: (a) current, (b) SOC, (c) voltage, and (d) temperature.

law with diffusion coefficients D^s_{eff},

$$\frac{\partial}{\partial t} c_s(z,t) = \frac{1}{z^2}\frac{\partial}{\partial z}\left[z^2 D^s_{\text{eff}} \frac{\partial}{\partial z} c_s(z,t)\right], \tag{18}$$

with boundary conditions

$$\frac{\partial}{\partial z} c_s(z,t)\bigg|_{z=0} = 0, \quad \frac{\partial}{\partial z} c_s(z,t)\bigg|_{z=R_s} = -\frac{j(z,t)}{D^s_{\text{eff}}}, \tag{19}$$

where z stands for the one-dimension spatial variable, t is time, $c_s(z,t)$ represents the concentration of solid particles, R_s is the radius of solid particles, and $j(z,t)$ is the ionic flux.

We define the state of charge (SOC) of the anode as

$$SOC(t) := \frac{1}{L_n c_s^{\max,n}} \int_0^{L_n} c_s(z,t)\mathrm{d}z, \tag{20}$$

where $c_s^{\max,n}$ is the maximum of $c_s(z,t)$ in the anode and L_n

Algorithm 2 Adaptive GP model-based RL

```
Apply static GP-based RL on an initial environment to obtain initial GP ($G_0$) and initial actor-critic networks (whose parameters are $\theta_0^\pi$ and $\theta_0^Q$, respectively);
for $n = 1$ to $N_{\text{cycles}}$ do
    Initialize a new agent with parameters as $\theta_n^\pi = \theta_0^\pi$ and $\theta_n^Q = \theta_0^Q$;
    for $j = 1$ to $N'$ do
        Simulate a complete process, update $\theta_n^\pi$ and $\theta_n^Q$ as in steps 6, 8, and 9 in Algorithm 1;
        Calculate second order difference to update differential GP ($G'_n$);
    end for
    Integrate $G_0$ and $G'_n$ to form a GP model ($G_n$) for the new environment;
    repeat
        $G_n$-based training as in steps 16-18 in Algorithm 1;
    until Convergence
end for
```

is the thickness. The voltage can be calculated as

$$V(t) = \Phi_s(0,t) - \Phi_s(L,t), \tag{21}$$

where $z = 0$ and $z = L$ correspond to the current collector at the cathode and anode sides, and $\Phi_s(z,t)$ is the solid overpotential at location z and time t. The temperature dynamics is modeled as

$$\begin{aligned}\rho C_p \frac{\partial}{\partial z} T(z,t) =& \frac{\partial}{\partial z}\left[\lambda \frac{\partial}{\partial z} T(z,t)\right] + Q_{\text{ohm}}(z,t) \\ &+ Q_{\text{rxn}}(z,t) + Q_{\text{rev}}(z,t),\end{aligned} \tag{22}$$

where ρ is the material density, C_p is the specific heat, λ is the thermal conductivity, and the terms Q_{rev}, Q_{rxn}, and Q_{ohm} stand for reversible, reaction, and ohmic heat sources, respectively [29]. Our study will be based on PETLION [30], an open-source, high-performance implementation of the PET model in Julia, as the battery simulator.

B. Minimum Time Battery Charging Problem for RL

Our objective is to minimize the charging time for batteries without violating operating constraints. The problem of minimizing charging time based on RL formulates as

$$\begin{aligned}\max_{I(t)} \quad & -t_f \\ \text{s.t.} \quad & \text{battery dynamics in (20)-(25),} \\ & I_{\min} \le I(t) \le I_{\max}, \\ & T(t) \le T_{\max}, V(t) \le V_{\max}, \\ & T(t_0) = T_0, V(t_0) = V_0, \\ & SOC(t_0) = SOC_0, SOC(t_f) = SOC_{\text{ref}},\end{aligned} \tag{23}$$

where t_0 and t_f are the initial time and final time of the charging process; T_0, V_0, and SOC_0 are the initial values of the temperature, voltage, and SOC, respectively. SOC_{ref} is the reference of SOC at which the charging is considered finished, $T_{\max}$ and $V_{\max}$ is the upper bounds for battery temperature and voltage, respectively. Note that in (23), to reduce the SEI formation, we explicitly include constraints of voltage and battery since their values are directly accessible in practice and also they are tightly related to the SEI. Although other variables such as voltage overpotential is an even better reflection of SEI, their values are not measurable in practice and thus it is not convenient to include them in the optimization (23) or the proposed RL framework. Here, we treat the dynamics (20)-(25) as a black-box. The details of solving (23) while maintaining the constraints using RL algorithms are elaborated in the next section.

V. Results and Discussion

A. Model-Based RL for Constrained Charging Policy

In this section, our objective is to assess the proposed model-based RL charging approach in the context of a dynamically changing environment (i.e., aging battery), in comparison with the model-free RL charging method [27]. Our goal is to optimize the charging policy that charges the battery from 20% state of charge (SOC) to 80% SOC in minimal time, while keeping the temperature and voltage within operational constraints.

The upper limits of temperature and voltage are $T_{\max} = 309\text{K}$ and $V_{\max} = 4.2\text{V}$, the ambient temperature is $T_{\text{amb}} = 298.15\text{K}$, and the charging currents are set within the range $[0.05\text{C}, 4\text{C}]$. The reason we set such temperature and voltage limits is because cell degrades faster at higher temperatures (e.g., ¿ 40^{o}C) and voltages (e.g., ¿ 4.1V) by causing accelerated consumption of lithium inventory and building of internal resistance [31]. Moreover, setting such limits can ensure that traditional single-phase porous electrode theory [32] is valid for Li-plating-based degradation in designing fast-charging protocols [32]. For the RL framework, the discounting factor is $\gamma = 0.99$, the learning rates of the critic and actor networks are $\eta_Q = 0.0001$ and $\eta_\pi = 0.001$, respectively, and the sampling time is $T = 30\text{s}$. V_0 is assumed to be a uniformly distributed random variable ranging from 3.2V to 3.8V, and T_0 is similarly a uniformly distributed random variable ranging from 18°C to 32°C.

In each episode, we simulate a charging process with the environment (battery) state defined as $\mathbf{s}_i = (SOC_i, V_i, T_i)$, actions defined as $\mathbf{a}_i = I_i$, where the subscript i denotes the i-th RL step in this episode, and instant reward defined as:

$$r_{i+1} = r_{i+1}^{SOC} + r_{i+1}^{t} + r_{i+1}^{V} + r_{i+1}^{T}, \tag{24}$$

where

$$r_{i+1}^{SOC} = 10 \times (SOC_{i+1} - SOC_i),$$
$$r_{i+1}^{t} = -0.01 \times (t_{i+1} - t_i),$$
$$r_{i+1}^{V} = \begin{cases} -2 \times (V_{i+1} - V_{\max}), & V_{i+1} > V_{\max} \\ 0, & V_{i+1} \le V_{\max} \end{cases}$$
$$r_{i+1}^{T} = \begin{cases} -(T_{i+1} - T_{\max}), & T_{i+1} > T_{\max} \\ 0. & T_{i+1} \le T_{\max} \end{cases}$$

The reward function is designed to encourage the battery to be charged to 80% SOC as fast as possible, while keeping the temperature and voltage within the operational constraints.

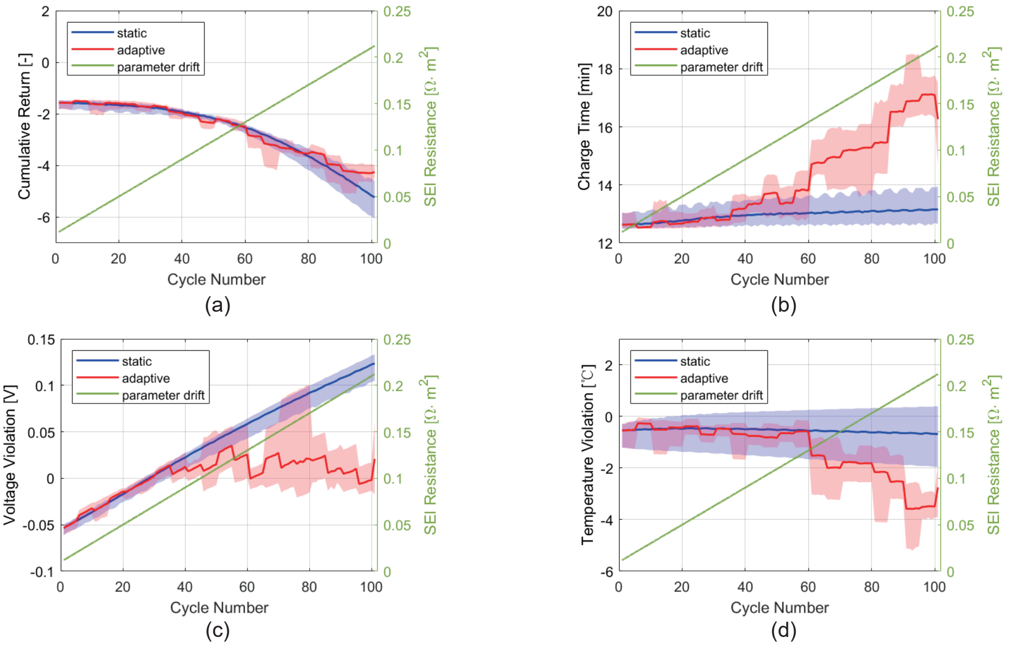


Fig. 4. Charging performance for an aging battery using the proposed adaptive model-based RL and the static model-based approaches at $T_{\text{amb}} = 25°$C: (a) cumulative return, (b) charge time, (c) voltage violation, and (d) temperature violation. Static policy is the optimal policy learned from the first charging cycle whose parameters are fixed as battery degrades, and parameter drift means the drifting of SEI resistance representing battery aging. The shaded region stands for the range between the maximum and minimum values of a variable, and the line stands for its mean value.

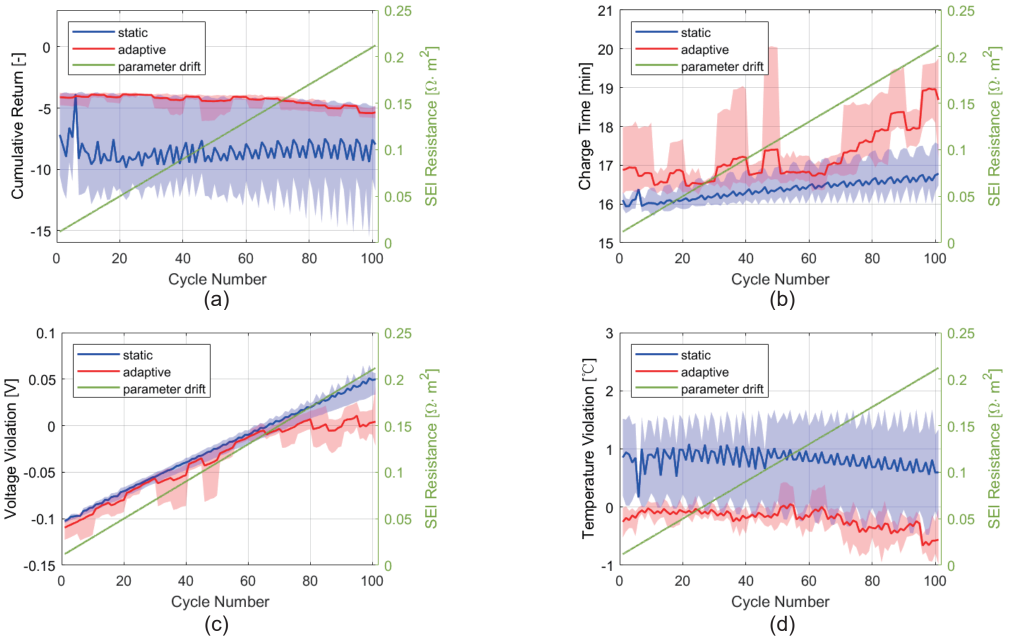


Fig. 5. Charging performance for an aging battery using the proposed adaptive model-based RL and the static model-based RL methods for the case of ambient temperature changed from $T_{\text{amb}} = 25°$C to $T_{\text{amb}} = 30°$C: (a) cumulative return, (b) charge time, (c) voltage violation, and (d) temperature violation. Static policy is the optimal policy learned from the scenario of ambient temperature at $T_{\text{amb}} = 25°$C and its parameters is fixed as ambient temperature changes, and parameter drift means the drifting of SEI resistance representing battery aging. The shaded region stands for the range between the maximum and minimum values of a variable, and the line stands for its mean value.

As mentioned in Section III, in the proposed model-based RL approach, the first 50 episodes are used to learn the parameters of the actor-critic network using the model-free manner, during which the inputs $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ and the outputs $\mathbf{y} = [\Delta_1, \Delta_2, \ldots, \Delta_n]$ of state transition are stored. A surrogate model of GP is then modeled for the environment, so that the actor-critic networks can continuously interact with the GP model instead of the environment for learning the optimal charging protocols.

For our case study, the state of environment is described as a three-dimensional state vector with three entries: state-of-charge, voltage and temperature. The action is described as a scalar of the charging current to apply in the following period of sampling time. The actor network is a multi-layer perceptron (MLP) with two hidden layers. Each hidden layer contains 20 nodes. The activation function is selected as ReLU. The input to the actor network is the state of environment (3 dimensions), and the output is the generated action by the actor to be applied to the environment (1 dimension). The critic network is also an MLP with two hidden layers with 100 nodes in the first layer and 75 nodes in the second layer, with ReLU as the activation function. The input to the critic network is the concatenation of state vector and action scalar (4 dimensions in total). Then the critic network returns the Q-value as an assessment of the previously deployed action (1 dimension).

The learning processes of the model-based RL charging method and the model-free RL method are shown in Fig. 2. The results show that both the RL learners possesse similar performance for the minimum-time charging problem, and both methods converge to similar optimal charging solutions within 300 episodes. However, the proposed model-based RL interacts with the real battery system only in first 50 episodes. Afterwards it interacts with the trained GP model instead during the subsequent training process. It means that the proposed model-based RL has a factor of 6 less expensive than the model-free method in terms of experimental cost for the fast charging design.

Fig. 2 (c)-(d) show the degree that voltage and temperature violate their respective constraints during learning, in which the violation score is computed as $\max_{0\le t\le t_f} V_t - V_{\max}$ and $\max_{0\le t\le t_f} T_t - T_{\max}$, respectively for each episode. Negative values indicate that the constraints are satisfied and positive ones indicate constraint violation. The constraint violation scores approach the boundary (i.e., zero) as the episodes increase, indicating that the RL approach learns not only the voltage and temperature constraints but also the optimal solution of charging policy (since the optimal solution is located around constraint boundaries). From Fig. 2b, it is observed that the minimum-charging time optimized by the RL methods is about 12.5 minutes. The corresponding optimal charging profile for applied current, SOC, cell voltage as well as cell temperature is displayed in Fig. 3.

B. Learning Adaptive Constrained Charging Policy for Dynamically Aging Battery

This section investigates the adaptability of the proposed adaptive model-based RL charging approach as battery degrades. Here the aging process of battery is represented by linearly increasing the value of SEI resistance as battery cycling (increase rate = $0.002\Omega \cdot \mathrm{m}^2$/cycle).

In terms of the adaptive model-based RL approach, we update the parameters of actor-critic based on interactions with the simulation environment for the first 30 episodes, and simultaneously train the differential GP as stated in Section III.B. After that, we combine the differential GP model and the GP model learned from the data of first cycle for predicting the state transition according to (19). The parameters of actor and critic networks are then updated based on the interaction with the GP models instead of the environment in the remaining 270 episodes. The same architecture of the actor and critic networks from the previous case study is used here.

The learning results of the static and adaptive model-based RL for each cycle at $T_{\text{amb}} = 25°\text{C}$ are shown in Fig. 4. As shown in Fig. 4 (b)-(c), the charging time and voltage violation of both static and adaptive policy increases as the battery degrades, which is a consequence of the increase in the SEI resistance. As shown in Fig. 4 (a), the rewards of both static and adaptive policy decrease as the degradation of battery, but the reasons for their drops are different. From Fig. 4 (b)-(d), the increase rate of charging time of static policy is relatively slow, and the static policy fails quickly as its voltage violation exceeds 0 after just 30 cycles. Therefore, the decrease of static reward is mainly from the violation of voltage constraint. The charging time of adaptive policy increases more rapidly than the static policy, but there's little constraint violations on both voltage and temperature as the degradation of battery. In short, the adaptive RL charging approach exhibits a better performance in handling the degradation constraints than the static RL method in the dynamically changing environment. One observation in Fig. 4 (d) is that the temperature constraint is satisfied with a large margin after 60 cycles, indicating a conservative policy. That is because for a battery, the voltage and temperature are inter-related with complex electrochemical relations. Thus, meeting the temperature constraint with a large margin as in Figure 4. (d) may be a prerequisite for ensuring the satisfaction of voltage constraints. In other words, if the temperature were kept close to the boundary after 60 cycles, the voltage would exceed the limit.

The ambient temperature T_{amb} is also altered to further evaluate the performance of the adaptive RL charging approach. The learning results of the static and adaptive model-based RL for each cycle at $T_{\text{amb}} = 30°\text{C}$ are displayed in Fig. 5. Fig. 5 shows that the charging time and voltage violation of both static and adaptive policy increases as the aging of battery, which is similar to Fig. 4. The reward of adaptive policy is higher than that of static policy, as shown in Fig. 5 (a). From Fig. 5 (b)-(d), although the charging time of the static policy is shorter, its temperature violation is significantly larger than that of adaptive policy, which causes more severe degradation to batteries.

It is worth mentioning that a difference between Fig. 4 and Fig. 5 is in the temperature violation. Fig. 5 shows that the charging policy provided by the static model-based RL violates the temperature constraints (temperature violation score is about $1°\text{C}$) throughout the whole process as battery degrades, which is mainly because the static policy is not adaptive to the change of ambient temperature. In contrast, the charging policy optimized by the adaptive model-based RL can maintain the constraint of temperature (its temperature violation score is around the boundary of $0°\text{C}$), as the ambient temperature changes from $25°\text{C}$ to $30°\text{C}$. Fig. 5 and Fig. 4 show that the adaptive RL approach performs well in the handling of both voltage and temperature constraints for the scenario of change on ambient temperature. As a compensation, the charging time of the adaptive policy increases (see Fig. 4 (b) and Fig. 5 (b)).

In summary, the charging policy optimized by static RL method would easily violate constraints as battery degrades. On the contrary, the proposed adaptive model-based RL charging approach can effectively prevent such violations and obtain superior performance of fast charging optimization for the changes in the battery environment.

VI. Conclusion

In this article, an adaptive model-based deep RL approach is proposed for the optimization of minimum-charging time policies for dynamically aging batteries with constraints on voltage and temperature for limiting battery degradation. The proposed approach is verified on a PET model-based battery simulator. The results show that a model-based RL is superior to a model-free one in terms of sample efficiency. Both methods converge to similar optimal solutions without degradation constraint violation in first 300 episodes of a single charging cycle, while interaction episodes are 300 for model-free RL method and 50 for model-based RL method, which means that the proposed approach can largely reduce the experimental cost of fast charging design. In addition, the adaptive model-based RL shows superior performance in handling the degradation constraints of voltage and temperature with high sample efficiency for the scenario of dynamically aging batteries. In terms of future work, the proposed RL strategy can be applied to include the optimization of battery design parameters such as the next-generation solid-state electrolyte chemistries to advance the development of high-performance batteries.

References

[1] M.A. Hannan et al., “State-of-the-art and energy management system of lithium-ion batteries in electric vehicle applications: Issues and recommendations,” *IEEE Access*, vol. 6, pp. 19362-19378, Mar. 2018.

[2] Z.P. Cano et al., “Batteries and fuel cells for emerging electric vehicle markets,” *Nature Energy*, vol. 3, no. 4, pp. 279-289, Apr. 2018.

[3] Y. Liu et al., “Challenges and opportunities towards fast-charging battery materials,” *Nature Energy*, vol. 4, no. 7, pp. 540-550, Jul. 2019.

[4] X. Yang et al., “Fast charging of lithium-ion batteries at all temperatures,” *Proceedings of the National Academy of Sciences*, vol. 115, no. 28, pp. 7266-7271, Jul. 2018.

[5] K.A. Severson et al., “Data-driven prediction of battery cycle life before capacity degradation,” *Nature Energy*, vol. 4, no. 5, pp. 383-391, May. 2019.

[6] S. Ahmed et al., “Enabling fast charging—A battery technology gap assessment,” *Journal of Power Sources*, vol. 367, pp. 250-262, Nov. 2017.

[7] R. Klein, N.A. Chaturvedi, J. Christensen, J. Ahmed, R. Findeisen, and A. Kojic, “Optimal charging strategies in lithium-ion battery,” *Proceedings of the 2011 American Control Conference*, San Francisco, CA, USA, Jun. 2011, pp. 382-387.

[8] B. Suthar et al., “Optimal charging profiles for mechanically constrained lithium-ion batteries,” *Physical Chemistry Chemical Physics*, vol. 16, no. 1, pp. 277-287, Sep. 2014.

[9] M.A. Xavier, *Efficient Strategies for Predictive Cell-Level Control of Lithium-Ion Batteries*, University of Colorado, Springs, 2016.

[10] B. Jiang and X. Wang, “Constrained bayesian optimization for minimum-time charging of lithium-ion batteries,” *IEEE Control Systems Letters*, vol. 6, pp. 1682-1687, Nov. 2021.

[11] S. Park et al., “Reinforcement learning-based fast charging control strategy for li-ion batteries,” *IEEE Conference on Control Technology and Applications (CCTA)*, Montreal, QC, Canada, Aug. 2020, pp. 100-107.

[12] J.R. Vázquez-Canteli and Z. Nagy, “Reinforcement learning for demand response: A review of algorithms and modeling techniques,” *Applied Energy*, vol. 235, pp. 1072-1089, Feb. 2019.

[13] V. Gullapalli, “A stochastic reinforcement learning algorithm for learning real-valued functions,” *Neural Networks*, vol. 3, no. 6, pp. 671-692, Jan. 1990.

[14] L. Zhao and M.I. Roh, “COLREGs-compliant multiship collision avoidance based on deep reinforcement learning,” *Ocean Engineering*, vol. 191, p. 106436, Nov. 2019.

[15] X. Ruan et al., “Mobile robot navigation based on deep reinforcement learning,” *2019 Chinese control and decision conference (CCDC)*, Nanchang, China, Jun. 2019, pp. 6174-6178.

[16] F. Bu and X. Wang, “A smart agriculture IoT system based on deep reinforcement learning,” *Future Generation Computer Systems*, vol. 99, pp. 500-507, Oct. 2019.

[17] T. Haarnoja et al., “Soft actor-critic algorithms and applications,” arXiv:1812.05905, Dec. 2018.

[18] C.G. Atkeson and J.C. Santamaria, “A comparison of direct and model-based reinforcement learning,” *Proceedings of International Conference on Robotics and Automation*, Albuquerque, NM, USA, Apr. 1997, pp. 3557-3564.

[19] R.S. Sutton and A.G. Barto, *Reinforcement learning: An introduction.* MIT Press, 2018.

[20] K. Lee et al., “Context-aware dynamics model for generalization in model-based reinforcement learning,” *International Conference on Machine Learning*, Nov. 2020, pp. 5757-5766.

[21] V. Pong et al., “Temporal difference models: Model-free deep RL for model-based control,” arXiv:1802.09081, Feb. 2018.

[22] M. Ahmadi et al., “Machine learning for high-throughput experimental exploration of metal halide perovskites,” *Joule*, vol. 5, no. 11, pp. 2797-2822, Nov. 2021.

[23] X. Yang and C. Wang, “Understanding the trilemma of fast charging, energy density and cycle life of lithium-ion batteries,” *Journal of Power Sources*, vol. 402, pp. 489-498, Oct. 2018.

[24] S. Zhang, K. Zhao, T. Zhu, and J. Li, “Electrochemomechanical degradation of high-capacity battery electrode materials,” *Progress in Materials Science*, vol. 89, pp. 479-521, Aug. 2017.

[25] B. Jiang et al., “Fast charging design for Lithium-ion batteries via Bayesian optimization,” *Applied Energy*, vol. 307, p. 118244, Feb. 2022.

[26] R. S. Sutton et al.,“Policy gradient methods for reinforcement learning with function approximation,” *Advances in Neural Information Processing Systems*, Nov. 1999. pp. 1057–1063.

[27] S. Park et al., “A deep reinforcement learning framework for fast charging of li-ion batteries,” *IEEE Transactions on Transportation Electrification*, vol. 8, no. 2, pp. 2770-2784, Jan. 2022.

[28] C. Zhu et al., “Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 23, no. 4, pp. 550-560, Dec. 1997.

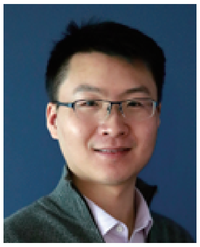
[29] T.F. Fuller, M. Doyle, and J. Newman, “Simulation and optimization of the dual lithium ion insertion cell,” *Journal of the Electrochemical Society*, vol. 141, no. 1, p. 1, 1994.

[30] M.D. Berliner et al., “Methods—PETLION: Open-source software for millisecond-scale porous electrode theory-based lithium-ion battery simulations,” *Journal of The Electrochemical Society*, vol. 168, no. 9, p. 090504, Sep. 2021.

[31] D.P. Finegan, A. Quinn, D.S. Wragg, A.M. Colclasure, X. Lu, C. Tan, T.M. Hennan, R. Jervis, D.J. Brett, S. Das, and T. Gao, “Spatial dynamics of lithiation and lithium plating during high-rate operation of graphite electrodes,” *Energy & Environmental Science*, 13(8): 2570-2584, 2020.

[32] M. Torchio, L. Magni, R.B. Gopaluni, R.D. Braatz, and D.M. Raimondo, “LIONSIMBA: A Matlab framework based on a finite volume model suitable for Li-ion battery design, simulation, and control,” *Journal of The Electrochemical Society*, 163: A1192-A1205, 2016.

Yuhan Hao is an undergraduate student in the Department of Automation at Tsinghua University, Beijing, China. His research interests include reinforcement learning and its application to battery fast charging design.



Qiugang Lu is an Assistant Professor in the Department of Chemical Engineering at the Texas Tech University. He received the B.Eng. in control science and engineering from Harbin Institute of Technology, China, in 2011, and the Ph.D. in chemical and biological engineering from the University of British Columbia, Canada, in 2018. Prior to this position, he was a Prognostics Engineer at the General Motors of Canada in Toronto from 2018 to 2019, and a Postdoctoral Research Associate at the University of Wisconsin-Madison from 2019 to 2020. His research interests include advanced process control, data analytics, system identification, and process monitoring. He has published over 30 articles in renowned journal and conferences. Dr. Lu is a recipient of Vanier Canada Graduate Scholarship (2015), Best Presenter Award at PACWEST Conference (2015), and Certificate of Service Award from the Journal of the Franklin Institute (2019).

Xizhe Wang is currently working toward the Ph.D. degree in control science and engineering in the Department of Automation at Tsinghua University. His research interests include Bayesian optimization and reinforcement learning with application to the design of next-generation battery technology.

Benben Jiang is an Assistant Professor in the Department of Automation at Tsinghua University. He received the B.Eng. in automation from Zhejiang University, China, in 2010 and the Ph.D. in control science and engineering from Tsinghua University, China, in 2015. Prior to his current position, he was a Postdoctoral Associate at the Massachusetts Institute of Technology (MIT) from 2016 to 2020. His research interests include computational energy intelligence, machine learning with application to advanced battery systems, and fault diagnosis for manufacturing processes. Dr. Jiang is a recipient of the 2016 Prize in Informatics/Computer Science from the Dimitris Chorafas Foundation, Switzerland. He was also awarded by the Department of Automation, Tsinghua University, for being a Promising Young Researcher, and he received the 2015 Outstanding Doctoral Dissertation from Tsinghua University.