A Deep Reinforcement Learning Framework for Fast Charging of Li-Ion Batteries

Saehong Park, *Member, IEEE*, Andrea Pozzi, Michael Whitmeyer, Hector Perez, Aaron Kandel, Geumbee Kim, Yohwan Choi, Won Tae Joe, Davide M. Raimondo, *Member, IEEE*, and Scott Moura, *Member, IEEE*

***Abstract*— One of the most crucial challenges faced by the Li-ion battery community concerns the search for the minimum time charging without damaging the cells. This goal can be achieved by solving a large-scale constrained optimal control problem, which relies on accurate electrochemical models. However, these models are limited by their high computational cost, as well as identifiability and observability issues. As an alternative, simple output-feedback algorithms can be employed, but their performance strictly depends on trial and error tuning. Moreover, particular techniques have to be adopted to handle safety constraints. With the aim of overcoming these limitations, we propose an optimal-charging procedure based on deep reinforcement learning. In particular, we focus on a policy gradient method to cope with continuous sets of states and actions. First, we assume full state measurements from the Doyle–Fuller–Newman (DFN) model, which is projected to a lower dimensional feature space via the principal component analysis. Subsequently, this assumption is removed, and only output measurements are considered as the agent observations. Finally, we show the adaptability of the proposed policy to changes in the environment's parameters. The results are compared with other methodologies presented in the literature, such as the reference governor and the proportional–integral–derivative approach.**

***Index Terms*— Actor–critic, approximate dynamic programming (ADP), electrochemical model (EM), fast charging, reinforcement learning (RL).**

NOMENCLATURE

A. Electrochemical–Thermal Model States, Inputs, and Outputs

$c_s^{\pm}$	Lithium concentration in the solid phase [mol/m^3].
c_e	Lithium concentration in the electrolyte phase [mol/m^3].
T	Cell temperature [K].
$\phi_s^{\pm}$	Solid electric potential [V].
ϕ_e	Electrolyte electric potential [V].
$i_e^{\pm}$	Ionic current [A/m^2].
$j_n^{\pm}$	Molar ion flux [mol/m^2-s].
$i_0^{\pm}$	Exchange current density [A/m^2].
$\eta^{\pm}$	Overpotential [V].
$c_{\text{ss}}^{\pm}$	Lithium concentration at the solid particle surface [mol/m^3].
$\theta^{\pm}$	Stoichiometry [-].
I	Applied current [A/m^2].
V	Terminal voltage [V].

B. Electrochemical–Thermal Model Parameters

$D_s^{\pm}$ and D_e	Diffusivity of solid and electrolyte phases [m^2/s].
t_c^0	Transference number [-].
$\varepsilon_s^{\pm}$ and ε_e	Volume fraction of solid and electrolyte phases [-].
F	Faraday's constant [C/mol].
$\sigma^{\pm}$	Conductivity of solid [1/Ω-m].
κ	Conductivity of electrolyte [1/Ω-m].
R	Universal gas constant [J/mol-K].
$f_{c/a}$	Mean molar activity coefficient in electrolyte [-].
$a^{\pm}$	Specific interfacial surface area [m^2/m^3].
α_a and α_c	Anodic and cathodic charge transfer coefficients [-].
$k^{\pm}$	Kinetic reaction rate [(A/m^2)(mol^3/mol)$^{(1+\alpha)}$].
$c_{s,\max}^{\pm}$	Maximum concentration of solid material [mol/m^3].
$U^{\pm}$	Open-circuit potential of solid material [V].
$R_f^{\pm}$	Solid–electrolyte interphase film resistance [Ω-m^2].
$R_s^{\pm}$	Particle radius in the solid phase [m].
L^j	Length of region $j \in \{-, \text{sep}, +\}$.
E_ψ	Activation energy [J/mol].
c_P	Heat capacity of the cell [J/(Kg-K)].
R_{th}	Thermal resistance [K/W].
m	Mass of cell [Kg].

Manuscript received August 13, 2021; revised November 4, 2021; accepted December 19, 2021. Date of publication January 4, 2022; date of current version April 20, 2022. This work was supported in part by the LG Chem Battery Innovative Contest and in part by the Italian Ministry for Research in the Framework of the 2017 Program for Research Projects of National Interest (PRIN) under Grant 2017YKXYXJ. An earlier version of this article was presented at the IEEE Conference on Control Technology and Applications (CCTA), Montreal, QC, Canada, August 24, 2020 [DOI: 10.1109/CCTA41146.2020.9206314]. *(Corresponding author: Saehong Park.)*

Saehong Park, Michael Whitmeyer, Hector Perez, Aaron Kandel, and Scott Moura are with the Energy, Controls and Applications Laboratory (eCAL), University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: sspark@berkeley.edu; mwhitmeyer@berkeley.edu; heperez@berkeley.edu; aaronkandel@berkeley.edu; smoura@berkeley.edu).

Andrea Pozzi and Davide M. Raimondo are with the Department of Industrial and Information Engineering, University of Pavia, 27100 Pavia, Italy (e-mail: andrea.pozzi03@universitadipavia.it; davide.raimondo@unipv.it).

Geumbee Kim, Yohwan Choi, and Won Tae Joe are with the BMS Advanced SW Project Team, LG Chem, Daejeon 305-738, South Korea (e-mail: gbeekim@lgchem.com; nicehwan@lgchem.com; wontaejoe@lgchem.com).

Digital Object Identifier 10.1109/TTE.2022.3140316

I. INTRODUCTION

LITHIUM-ION batteries are crucial technologies for electrified transportation, clean power systems, and consumer

electronics. Although Li-ion batteries exhibit promising features in terms of energy and power density, they still present limited capacity and long charging time [1]. While the former is mostly related to the battery chemistry, materials, and design, the latter depends on the employed charging strategy. Within this context, the tradeoff between fast charging and aging has to be taken into account. In fact, charging time reductions can be easily achieved by using aggressive current profiles, which, in turn, may lead to severe battery degradation effects, such as solid–electrolyte interphase (SEI) growth and lithium plating deposition. Consequently, safety constraints must be enforced in order to prevent possible thermal runaway and overcharge.

The most common charging procedure for Li-ion batteries is the well-known constant-current constant-voltage (CC-CV) method. This is employed in the industry due to its ability to provide reasonable performance with a relatively simple implementation [2]. However, such a simple charging algorithm is often based on excessively conservative constraints that reduce the probability of safety hazards at the expense of higher charging times. Therefore, it does not constitute an optimal policy for the problem that we aim to solve—at least not in all cases. For these reasons, several advanced battery management strategies have been employed. In particular, we can classify them as: 1) model-based strategies and 2) model-free strategies. The former seeks to find an optimal input trajectory based on a specified battery model, while the latter interacts directly with the battery [denoted as the "environment" in the language of reinforcement learning (RL)].

The use of mathematical models for battery control is a large topic in the literature. Equivalent circuit models are simple, intuitive, and mimic the battery behaviors through lumped electrical parameters, which can be easily identified [3]. Electrochemical models (EMs) exhibit higher accuracy than equivalent circuit models and the ability to describe internal battery phenomena from the perspective of electrochemistry and, therefore, are usually preferred for simulation purposes [4]. There are a number of studies to investigate the health-aware fast charging strategy for EMs. Klein *et al.* [5] formulate a minimum-time charging problem with health-related constraints and use nonlinear model predictive control. Similarly, Perez *et al.* [6] propose a reference governor approach to solve a minimum-time charging problem, and Torchio *et al.* [7] propose a quadratic dynamic matrix control formulation to design an optimal charging strategy for real-time model predictive control. Zou *et al.* [8] synthesize a state estimation and model predictive control scheme for a reduced electrochemical–thermal model, in order to design a health-aware fast charging strategy. The problem is formulated as a linear time-varying model predictive control scheme, with a moving horizon state estimation framework. In the context of aging mechanism for EMs, Perez *et al.* [9] studied the tradeoff between charging speed and degradation, based on an electrothermal-aging model. Pozzi *et al.* [10] minimize the film layer growth of an EM by formulating shrinking-horizon nonlinear model predictive control. The multiobjective optimal charging problem is considered for EM in [11] where the fast-charging strategy is characterized by three charging stages considering the fact of charging time, temperature rising, and charging loss. However, the exploitation of model-based charging procedures has to face some crucial challenges.

1) Every model is subject to uncertainties and modeling mismatches, which affects its accuracy. Since the controller's performance depends on the model accuracy, a proper parameter identification procedure has to be conducted based on experimentally collected data. In the case of EMs, there are typically dozens of parameters to be identified. This motivates sophisticated optimally designed experiments for parameter estimation [12], [13].
2) EMs usually consist of a large number of states, thus leading to a large-scale optimization problem. Moreover, most states are not measurable in a realistic scenario, and therefore, the presence of an observer is required to reconstruct the full state information from the available measurements [14].
3) The model parameters drift as the battery ages. It is important to notice that none of the model-based strategies proposed in the literature considers the adaptability of the control strategy to variations in the parameters.

In order to overcome all the limitations of the model-based approach, there has been a substantial effort in the literature to design fast-charging strategies that do not rely on a mathematical model. Some of them rely on rule-based adaptation of the CC-CV protocol [15]–[17]. Yin *et al.* [18] propose a charging algorithm that incorporates CC-CV charging profile and battery health estimation using an extended Kalman filter, where the magnitude of CC and the threshold for CV are updated from the estimator. Attia *et al.* [19] propose an optimal design of fast-charging procedures relying on machine learning. In particular, the current profile is parameterized by six steps of 10 min each, and the Bayesian optimization is used to select the optimal sequence, which maximizes battery cycle life. Patnaik *et al.* [20] propose closed-loop charging techniques called CC constant-temperature CV (CC-CT-CV), where a rule-based proportional–integral–derivative (PID) controller is employed. This closed-loop charging strategy constitutes an output-feedback control law, which enables the CC-CV protocol to consider temperature constraints. It is important to notice that the difficulty of the observability and identifiability is no longer an issue as this strategy relies only on the output measurements. The main issues of this output-based strategy are represented by the fact that: 1) the optimality of the resulting charging policy is no longer guaranteed; 2) the controller gain should be obtained by trial and error; and 3) the controller does not adapt to parameter changes.

All these challenges can be addressed by using a charging procedure based on the RL framework [21]. An RL framework consists of an agent (the battery management system), which interacts with the environment (the battery) by taking specific actions (the applied current) according to the environment configuration (charging time). The main idea is that that the agent learns the feedback control policy directly from interactions with the environment, namely, observations of the reward and state. The control policy is iteratively updated

to maximize the expected long-term reward. Notice that the reward has to be properly designed, so the agent learns how to accomplish the required task. Most RL algorithms can be classified into two different groups: tabular methods, e.g., Q-learning, state-action-reward-state-action (SARSA), and approximate solutions methods, which is also called approximate dynamic programming (ADP). While the former performs well only in the presence of small and discrete sets of actions and states, the latter can be used even with continuous state and action spaces, thus solving the so-called "curse of dimensionality." On the other hand, the convergence of the former is proven under mild assumptions, while no proof of convergence exists for the approximate methods in the general case. The recent success in several applications of RL using deep neural networks as function approximators has greatly increased expectations in the scientific community [22]–[25]. From a control systems perspective, the design of RL algorithms involves the computation of feedback control laws for dynamical systems via optimal adaptive control methods [26]. RL can be regarded as an indirect adaptive controller, wherein the parameters of the value function are estimated, and then, the controller is improved based on the estimated value function.

In this article, a fast-charging strategy subject to safety constraints, using a deep RL framework, is proposed as an extension of the authors' previous work [27]. While, in such previous work, only a proof of concept has been proposed in order to assess the applicability of RL to the context of lithium-ion battery management, a more sophisticated framework is here considered with the aim of developing an RL-based BMS from a control engineering perspective. As the first contribution, a physics-based model simulator is considered as the real plant in order to accurately represent the internal cell phenomena (such as aging dynamics). The electrochemical parameters are directly measured by electrochemists from the battery manufacturer. It is evident that such an accurate model cannot be used as the model for the controller design due to its high computational burden and its lack of observability and identifiability, thus further motivating the use of RL as a model-free control algorithm. As in [27], two control schemes are implemented: the first one considers full states accessibility, while the second one is based on the more realistic assumption that only the battery outputs are available. Finally, a significant contribution of this work is the development of an RL algorithm, which is able to adapt its action as the battery degrades with aging. In particular, a simulation scenario in which the cell parameters are drifted in time is considered. The results highlight the ability of RL to adapt to the environment changes by adjusting its parameters and, therefore, guarantee safety constraints satisfaction. Among RL algorithms, deep deterministic policy gradient (DDPG) [28] is adopted for RL algorithm. DDPG is an actor–critic method that deals with continuous state and action spaces. The safety constraints are considered soft constraints where the agent (controller) receives penalties in the reward function in the case of violation.

We summarize our novel contributions to the relevant literature as follows. First, RL is adopted to overcome the challenges of the fast-charging problem in battery management systems: computational complexity, observability, and adaptation. To mitigate the computational complexity associated with feedback controller design for a high-dimensional state space, we project the state onto a lower dimensional feature space via the principal component analysis (PCA). This approach, however, requires a state observer to estimate the states in a real-world application, which motivates an output feedback controller using voltage, temperature, and current measurements only. The validities of the state- and output-feedback controllers are tested in simulation and compared to other existing charging algorithms, namely, the reference governor and CC-CT-CV. Finally, we show that RL policies are capable of adaptation when the battery parameters are changing over the cycles. This demonstrates that the use of the proposed methodology enables adaptation to uncertain and drifting parameters, which is a distinct advantage over other conventional approaches.

This article is organized as follows. Section II briefly presents the RL approach. Section III describes the battery model we are interested in. Section V exhibits control problem formulation for fast charging. Section VI presents simulation results and discussion for the proposed framework. In Section VII, we summarize our work and provide perspectives on the future direction.

II. Reinforcement Learning Framework

A. Markov Decision Process, Policy, and Value Functions

In the following, we briefly review the Markov decision process (MDP) to provide the critical background. A thorough exposition can be found in [29] and [30]. In the MDP setting, we seek the best policy that maximizes the total rewards received from the environment, E (i.e., the plant). At each time step $t \in \mathbb{R}^+$, the environment conditions are described by a state vector, $\boldsymbol{s}_t \in \mathcal{S}$, where $\mathcal{S}$ is the state space, while the control policy picks an action $\boldsymbol{a}_t \in \mathcal{A}$, with $\mathcal{A}$ being the action space, which is based on the observation of the state $\boldsymbol{s}_t$. The action is, therefore, applied to the environment, whose state evolves to $\boldsymbol{s}_{t+1} \in \mathcal{S}$, according to the state-transition probability $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$, and the agent receives a scalar reward $\boldsymbol{r}_{t+1} = r(\boldsymbol{s}_t, \boldsymbol{a}_t)$. The policy is represented by π, which maps the state to the action and can be either deterministic or stochastic. The total discounted reward from time t onward can be expressed as

$$R_t = \sum_{k=0}^{\infty} \gamma^k r(\boldsymbol{s}_{t+k}, \boldsymbol{a}_{t+k}) \tag{1}$$

where $\gamma \in [0, 1]$ is the discounting factor.

The *state value function*, $V^\pi(\boldsymbol{s}_t)$, is the expected total discounted reward starting from state $\boldsymbol{s}_t$. In the controls' community, this is sometimes called the cost-to-go or reward-to-go. Importantly, note that the value function depends on the control policy. If the agent uses a given policy π to select actions starting from the state $\boldsymbol{s}_t$, the corresponding value function is given by

$$V^\pi(\boldsymbol{s}_t) = \mathbb{E}\big[R_t \mid \boldsymbol{s}_t\big]. \tag{2}$$

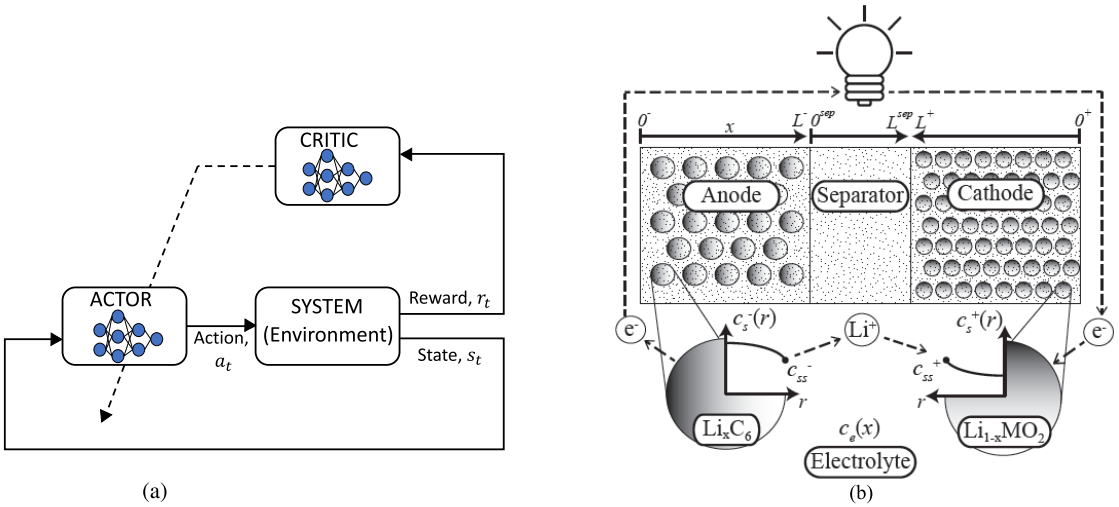


Fig. 1. RL framework for battery fast charging: (a) actor–critic scheme and (b) EM for the environment (schematic of first principles EM).

Then, the optimal policy π^* is the policy that corresponds to the maximum value $V^*(\boldsymbol{s}_t)$ of the value function

$$\pi^* = \arg\max_{\pi} V^{\pi}(\boldsymbol{s}_t). \tag{3}$$

The solution of (3) is pursued by those methods that follow the dynamic programming (DP) paradigm. Such a paradigm, however, assumes perfect knowledge of the environment E (i.e., the state-transition probability and the reward function).

The next definition, known as the "Q-function," is fundamental since it enables the concept of model-free RL. Consider the *state–action value function*, $Q^{\pi}(\boldsymbol{s}_t, \boldsymbol{a}_t)$, which is a function of the state–action pair that returns a real value. This corresponds to the expected total discounted reward when the action $\boldsymbol{a}_t$ is taken in state $\boldsymbol{s}_t$, and then, the policy π is followed henceforth. Mathematically,

$$Q^{\pi}(\boldsymbol{s}_t, \boldsymbol{a}_t) = \mathbb{E}\Big[R_t \mid \boldsymbol{s}_t, \boldsymbol{a}_t\Big]. \tag{4}$$

The optimal Q-function is given by

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

and represents the expected total discounted reward received by an agent that starts in $\boldsymbol{s}_t$, picks (possibly non-optimal) action $\boldsymbol{a}_t$, and then behaves optimally afterward. Since $V^*(\boldsymbol{s}_t)$ is the maximum expected total discounted reward starting from state $\boldsymbol{s}_t$, it will also be the maximum of $Q^*(\boldsymbol{s}_t, \boldsymbol{a}_t)$ over all possible actions $\boldsymbol{a}_t \in \mathcal{A}$

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

If the optimal Q-function is known, then the optimal action $\boldsymbol{a}_t^*$ can be extracted by choosing the action $\boldsymbol{a}_t$ that maximizes $Q^*(\boldsymbol{s}_t, \boldsymbol{a}_t)$ for state $\boldsymbol{s}_t$ (i.e., the optimal policy π^* is retrieved)

$$\boldsymbol{a}_t^* = \arg\max_{\boldsymbol{a}_t \in \mathcal{A}} Q^*(\boldsymbol{s}_t, \boldsymbol{a}_t) \tag{7}$$

without requiring the knowledge of the environment dynamics.

B. Actor–Critic

Actor–critic is an ADP method that solves DP heuristically, as depicted in Fig. 1(a). The actor–critic is a policy gradient approach, i.e., it seeks to model and optimize the policy directly. Importantly, the actor–critic approach allows for continuous state/action spaces by using a function approximator, e.g., a neural network. In RL, as well as in DP, the action is taken by a *policy* to maximize the expected total discounted reward. By following a given policy and processing the rewards, one should estimate the expected return given states from the *value function*. In the actor–critic approach, the *actor* improves the policy based on the value function that is estimated by the *critic*. We specifically focus on the policy gradient-based actor–critic algorithm in this work, and, in particular, the DDPG [28]. This algorithm is an extension of deep Q-network (DQN) [22] to continuous actions, maintaining the importance of features, such as: 1) random sampling from a replay buffer where tuples are saved and 2) the presence of target networks for stabilizing the learning process. The algorithm is characterized by the parameterization of both the critic, $Q(\boldsymbol{s}_t, \boldsymbol{a}_t|\boldsymbol{\theta}^Q)$, and the actor, $\pi(\boldsymbol{s}_t|\boldsymbol{\theta}^{\pi})$, as a deep neural network. In addition, the correspondent target networks are defined as $Q'(\boldsymbol{s}_t, \boldsymbol{a}_t|\boldsymbol{\theta}^{Q'})$ and $\pi'(\boldsymbol{s}_t|\boldsymbol{\theta}^{\pi'})$, where the parameters $\theta^{Q'}$ and $\theta^{\pi'}$ are slowly updated in order to track and filter the ones of the actual network (θ^Q and θ^{π}), thus reducing the chattering due to the learning process and enhancing its convergence. In the next paragraphs, the concepts of critic and actor are explained in detail. Both the actual and target parameters are initialized to the random vectors θ_0^{π} and θ_0^Q.

1) Critic: The role of the critic is to evaluate the current policy prescribed by the actor. The action is taken as the sum of the actor network output, which is a function of the environment states, and an exploration noise $\mathcal{N}_t$, namely,

$$\boldsymbol{a}_t = \pi(\boldsymbol{s}_t|\boldsymbol{\theta}^{\pi}) + \mathcal{N}_t \tag{8}$$

where the exploration noise is a random variable, for instance an Ornstein–Uhlenbeck process is considered in this article. After applying the action, we observe the reward $\boldsymbol{r}_{t+1}$ and the next state $\boldsymbol{s}_{t+1}$. For each time step t, the tuple $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_{t+1}, \boldsymbol{s}_{t+1})$

is stored in the replay buffer memory. As soon as the number of tuples stored in the memory reaches a default threshold N, at each time step, a random minibatch of N transitions is sampled from the buffer, and for each of them, we set

$$y_i = r_{i+1} + \gamma\, Q'\Big(\boldsymbol{s}_{i+1}, \pi'\Big(\boldsymbol{s}_{i+1}|\boldsymbol{\theta}^{\pi'}\Big)|\boldsymbol{\theta}^{Q'}\Big) \quad i = 1, \ldots, N \quad (9)$$

where superscript $'$ denotes the target network, whose parameters are slowly updated in (15). The minibatch, which is randomly extracted at each time step from the buffer if enough tuples are stored in the memory, is exploited for updating the networks. In particular, the critic is updated to minimize the loss function $\mathcal{L}(\boldsymbol{\theta}^Q)$

$$\mathcal{L}(\boldsymbol{\theta}^Q) = \frac{1}{N}\sum_i \big(y_i - Q(\boldsymbol{s}_i, a_i|\boldsymbol{\theta}^Q)\big)^2 \quad (10)$$

$$\boldsymbol{\theta}^Q_{k+1} = \boldsymbol{\theta}^Q_k - \eta_Q \nabla_{\boldsymbol{\theta}^Q} \mathcal{L}\big(\boldsymbol{\theta}^Q\big) \quad (11)$$

where index-k denotes the gradient descent algorithm iterates and η_Q denotes the learning rate of the critic network. Note that the subscript k in the network parameters $\boldsymbol{\theta}^Q_k$, $\boldsymbol{\theta}^\pi_k$, $\boldsymbol{\theta}^{Q'}_k$, and $\boldsymbol{\theta}^{\pi'}_k$ is omitted when clear from the context.

2) Actor: The parameters of the actor network are updated in order to maximize the cumulative expected reward $V^\pi(\boldsymbol{s}_t)$. In this paragraph, we refer to the cumulative reward with the variable $\mathcal{J}(\theta^\pi)$, in order to highlight its dependence on the actor parameterization. The update of the actor parameters is done as follows:

$$\boldsymbol{\theta}^\pi_{k+1} = \boldsymbol{\theta}^\pi_k + \eta_\pi \nabla_{\boldsymbol{\theta}^\pi} \mathcal{J}(\theta^\pi) \quad (12)$$

where index-k denotes the gradient ascent algorithm iterates and η_π denotes the learning rate of the actor network. Notice that, according to the proof in [31], the policy gradient in (12) can be expressed as

$$\nabla_{\boldsymbol{\theta}^\pi} \mathcal{J}(\theta^\pi) \approx \mathbb{E}\big[\nabla_{\boldsymbol{a}} Q(\boldsymbol{s}_t, \boldsymbol{a}_t|\boldsymbol{\theta}^Q) \nabla_{\boldsymbol{\theta}^\pi} \pi(\boldsymbol{s}_t|\boldsymbol{\theta}^\pi)\big] \quad (13)$$

which is then approximated by samples as follows:

$$\nabla_{\boldsymbol{\theta}^\pi} \mathcal{J}(\theta^\pi) \approx \frac{1}{N}\sum_i \big[\nabla_{\boldsymbol{a}} Q(\boldsymbol{s}_i, \boldsymbol{a}_i|\boldsymbol{\theta}^Q) \nabla_{\boldsymbol{\theta}^\pi} \pi(\boldsymbol{s}_i|\boldsymbol{\theta}^\pi)\big]. \quad (14)$$

Once the parameters of critic and actor network given samples are updated, then the target networks are also updated as follows:

$$\begin{aligned} \boldsymbol{\theta}^{Q'} &\leftarrow \tau\boldsymbol{\theta}^Q + (1-\tau)\boldsymbol{\theta}^{Q'} \\ \boldsymbol{\theta}^{\pi'} &\leftarrow \tau\boldsymbol{\theta}^\pi + (1-\tau)\boldsymbol{\theta}^{\pi'} \end{aligned} \quad (15)$$

where τ is the level of "soft-update." Equation (15) improves the stability of the learning procedure. Note that the convergence is no longer guaranteed, in general, when a value function approximator is used. Since the convergence of the critic network is not guaranteed, it is important to note that these target networks should update slowly to avoid divergence. Thus, one should choose a small value of τ. This is a challenging point when the action space becomes continuous, unlike tabular Q-learning. Finally, the actor and critic networks are trained from the database, called *replay buffer*, where the pair of (s_t, a_t, r_t, s_{t+1}) is stored by interacting with the environment. The RL model gets updated by using a stochastic gradient descent algorithm from randomly selected samples in the replay buffer.

III. Electrochemical–Thermal Model

Next, we review the mathematical battery model used for this study. Insights on battery cell design can be obtained via high fidelity battery models that allow for an assessment of the impact of physical parameters on battery performance. A mathematical model is built on the porous electrode theory where Li-ions are intercalated in spherical particles in the negative/positive electrodes. During charging, the Li-ions in the positive electrode are deintercalated, solved into the electrolyte, and then diffused to the negative electrode passing through the separator. Fig. 1 describes this process and shows the computation domain of the present model. We consider the Doyle–Fuller–Newman (DFN) model to predict the evolution of lithium concentration in the solid $c_s^\pm(x, r, t)$, lithium concentration in the electrolyte $c_e(x, t)$, solid electric potential $\phi_s^\pm(x, t)$, electrolyte electric potential $\phi_e(x, t)$, ionic current $i_e^\pm(x, t)$, molar ion fluxes $j_n^\pm(x, t)$, and battery temperature $T(t)$. The finite element is constructed in the x-direction, including negative electrode, separator, and positive electrode. Each finite element has active particles where spherical lithium intercalation occurs. The detailed symbols are defined in the Nomenclature. The governing equations are given by

$$\frac{\partial c_s^\pm}{\partial t}(x, r, t) = \frac{1}{r^2}\frac{\partial}{\partial r}\left[D_s^\pm r^2 \frac{\partial c_s^\pm}{\partial r}(x, r, t)\right] \quad (16)$$

$$\varepsilon_e^j \frac{\partial c_e^j}{\partial t}(x, t) = \frac{\partial}{\partial x}\left[D_e^{\text{eff}}(c_e^j)\frac{\partial c_e^j}{\partial x}(x, t) + \frac{1 - t_c^0}{F} i_e^j(x, t)\right] \quad (17)$$

$$mc_P \frac{dT}{dt}(t) = \frac{1}{R_{\text{th}}}[T_{\text{amb}} - T(t)] + \dot{Q} \quad (18)$$

for $j \in \{-, \text{sep}+\}$. $\dot{Q}$ is the rate of heat transferred to the system [32], defined as

$$\begin{aligned} \dot{Q} = I(t)\big[U^+(t) - U^-(t) - V(t)\big] \\ - I(t)T(t)\frac{\partial}{\partial T}\big[U^+(t) - U^-(t)\big] \end{aligned} \quad (19)$$

and the algebraic equations of the model are given by

$$\sigma^{\text{eff},\pm} \cdot \frac{\partial \phi_s^\pm}{\partial x}(x, t) = i_e^\pm(x, t) - I(t) \quad (20)$$

$$\begin{aligned} \kappa^{\text{eff}}(c_e) \cdot \frac{\partial \phi_e}{\partial x}(x, t) = -i_e^\pm(x, t) + \kappa^{\text{eff}}(c_e) \cdot \frac{2RT}{F}\big(1 - t_c^0\big) \\ \times\left(1 + \frac{d\ln f_{c/a}}{d\ln c_e}(x, t)\right)\frac{\partial \ln c_e}{\partial x}(x, t) \end{aligned} \quad (21)$$

$$\frac{\partial i_e^\pm}{\partial x}(x, t) = a^\pm F j_n^\pm(x, t) \quad (22)$$

$$j_n^\pm(x, t) = \frac{1}{F} i_0^\pm(x, t)\Big[e^{\frac{\alpha_a F}{RT}\eta^\pm(x,t)} - e^{-\frac{\alpha_c F}{RT}\eta^\pm(x,t)}\Big] \quad (23)$$

$$\begin{aligned} i_0^\pm(x, t) = k^\pm\big[c_{\text{ss}}^\pm(x, t)\big]^{\alpha_c} \\ \times\big[c_e(x, t)\big(c_{s,\max}^\pm - c_{\text{ss}}^\pm(x, t)\big)\big]^{\alpha_a} \end{aligned} \quad (24)$$

$$\begin{aligned} \eta^\pm(x, t) = \phi_s^\pm(x, t) - \phi_e(x, t) - U^\pm(c_{\text{ss}}^\pm(x, t)) \\ - F R_f^\pm j_n^\pm(x, t) \end{aligned} \quad (25)$$

$$c_{\text{ss}}^\pm(x, t) = c_s^\pm(x, R_s^\pm, t) \quad (26)$$

where $D_e^{\text{eff}} = D_e(c_e) \cdot (\varepsilon_e^j)^{\text{brug}}$, $\sigma^{\text{eff}} = \sigma \cdot (\varepsilon_s^j + \varepsilon_f^j)^{\text{brug}}$, and $\kappa^{\text{eff}} = \kappa(c_e) \cdot (\varepsilon_e^j)^{\text{brug}}$ are the effective electrolyte diffusivity, effective solid conductivity, and effective electrolyte conductivity given by the Bruggeman relationship. The boundary conditions for solid-phase diffusion PDE (16) are

$$\frac{\partial c_s^{\pm}}{\partial r}(x, 0, t) = 0 \tag{27}$$

$$\frac{\partial c_s^{\pm}}{\partial r}(x, R_s^{\pm}, t) = -\frac{1}{D_s^{\pm}} j_n^{\pm}(x, t). \tag{28}$$

The boundary conditions for the electrolyte-phase diffusion PDE (17) are given by

$$\frac{\partial c_e^-}{\partial x}(0^-, t) = \frac{\partial c_e^+}{\partial x}(0^+, t) = 0 \tag{29}$$

$$\varepsilon_e^- D_e(L^-) \frac{\partial c_e^-}{\partial x}(L^-, t) = \varepsilon_e^{\text{sep}} D_e(0^{\text{sep}}) \frac{\partial c_e^{\text{sep}}}{\partial x}(0^{\text{sep}}, t) \tag{30}$$

$$\varepsilon_e^{\text{sep}} D_e(L^{\text{sep}}) \frac{\partial c_e^{\text{sep}}}{\partial x}(L^{\text{sep}}, t) = \varepsilon_e^+ D_e(L^+) \frac{\partial c_e^+}{\partial x}(L^+, t) \tag{31}$$

$$c_e(L^-, t) = c_e(0^{\text{sep}}, t) \tag{32}$$

$$c_e(L^{\text{sep}}, t) = c_e(L^+, t). \tag{33}$$

The boundary conditions for the electrolyte-phase potential ODE (21) are given by

$$\phi_e(0^-, t) = 0 \tag{34}$$

$$\phi_e(L^-, t) = \phi_e(0^{\text{sep}}, t) \tag{35}$$

$$\phi_e(L^{\text{sep}}, t) = \phi_e(L^+, t). \tag{36}$$

The boundary conditions for the ionic current ODE (22) are given by

$$i_e^-(0^-, t) = i_e^+(0^+, t) = 0. \tag{37}$$

Note that $i_e(x, t) = I(t)$ for $x \in [0^{\text{sep}}, L^{\text{sep}}]$. In addition, the parameters, $D_s^{\pm}$, D_e, κ_e, and $k^{\pm}$, vary with temperature via the Arrhenius relationship

$$\psi = \psi_{\text{ref}} \exp\left[\frac{E_\phi}{R}\left(\frac{1}{T} - \frac{1}{T_{\text{ref}}}\right)\right] \tag{38}$$

where ψ represents a temperature dependent parameter, E_ψ is the activation energy, and ψ_{ref} is the reference parameter value at the room temperature. The model input is the applied current density $I(t)$ [A/m^2], and the output is the voltage measured across the current collectors

$$V(t) = \phi_s^+(0^+, t) - \phi_s^-(0^-, t). \tag{39}$$

The level of charge in the cell is defined by the bulk state of charge (SOC) of the negative electrode, namely,

$$\text{SOC}^-(t) = \int_0^{L^-} \frac{\bar{c}_s^-(x, t)}{c_{s,\max}^-(\theta_{100\%}^- - \theta_{0\%}^-)L^-} dx \tag{40}$$

where $\bar{c}_s^-$ represents the volume averaged of a particle in the solid phase defined as

$$\bar{c}_s^-(x, t) = \frac{3}{(R_s^-)^3} \int_0^{R_s^-} r^2 c_s^-(r, t) dr. \tag{41}$$

The main battery degradation mechanism, i.e., lithium plating [33]–[35], is related to the side reaction overpotential η_{sr}, which is defined as

$$\eta_{\text{sr}}(x, t) = \phi_s^-(x, t) - \phi_e^-(x, t) - U_{\text{sr}} \tag{42}$$

where U_{sr} denotes the equilibrium potential of the side reaction and is assumed to be zero [35]. A complete description of the model equations and notation can be found in [32] and [33]. Given the mathematical structure of the model, which contains linear PDEs (16), quasilinear PDEs (17), ODEs in space (20)–(22), and nonlinear algebraic constraints (23)–(25), it is possible to obtain nonlinear differential algebraic equations (DAEs) after a model discretization via suitable numerical methods, e.g., finite differences, Padé approximation, and spectral methods [36], [37]

$$\dot{\mathbf{x}} = \boldsymbol{f}(\mathbf{x}, \mathbf{z}, u), \quad \mathbf{x}(t_0) = \mathbf{x_0} \tag{43}$$

$$\mathbf{0} = \boldsymbol{g}(\mathbf{x}, \mathbf{z}, u), \quad \mathbf{z}(t_0) = \mathbf{z_0} \tag{44}$$

$$\mathbf{y} = \boldsymbol{h}(\mathbf{x}, \mathbf{z}, u) \tag{45}$$

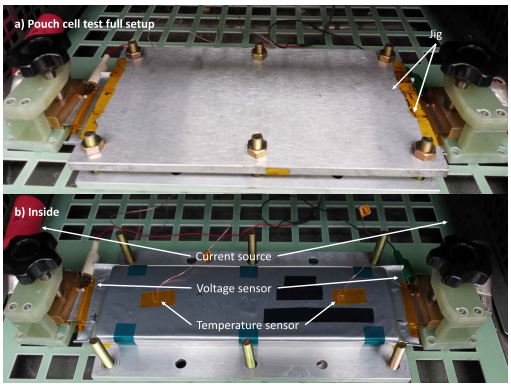
where $\mathbf{x} = [c_s^-, c_s^+, c_e, T]^T \in \mathbb{R}^{n_x}$ as state vectors, $\mathbf{z} = [\phi_s^-, \phi_s^+, i_e^-, i_e^+, \phi_e, j_n^-, j_p^+]^T \in \mathbb{R}^{n_z}$ as algebraic variables, $\mathbf{y} = V(t)$ as output variable defined in (39), and u is equivalent to the applied input current, $I(t)$, as shown in (20). The system is a semiexplicit DAE of index 1 as $\left[\partial \boldsymbol{g}/\partial \mathbf{z}\right]^{-1}$ exists. To simulate the model, we choose the sample time as $\delta t = 30s$.

IV. Experimental Model Validation

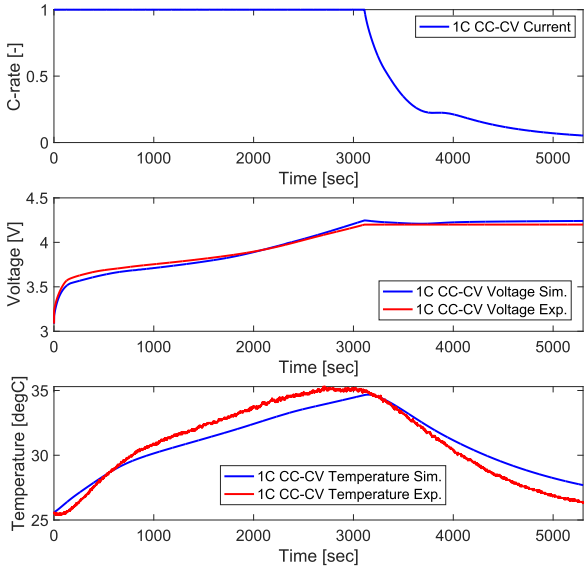
It is well known that a proper experimental validation phase is required in order to assess the model's accuracy in describing the cell behavior. Within this context, the use of an accurate digital twin of the electrochemical cell is fundamental since it allows the researchers to test the proposed control algorithms directly in simulation and, therefore, avoid time-consuming experiments on the real battery. In this section, we assess the level of confidence of the electrochemical thermal model presented in Section III. The electrochemical parameters of the cell (a graphite anode and LiNiMnCoO2 cathode chemistry) are experimentally measured by the cell manufacturer. The details of electrochemical parameters can be found in [13].

The experimental setup for model validation is depicted in Fig. 2(a). The jig maintains the uniform pressure to the pouch cell, while temperature and voltage sensors are placed inside the jig for measurements. To validate the thermal dynamics (18) accurately, we take the average of two temperature measurements. The comparison of simulation and experiment is plotted in Fig. 2(b). V, I, and T for the 1-C CC-CV charging protocol are obtained from the experiment and then plug-in to the EM.

The model can be validated by comparing various profiles, such as driving cycles, pulse inputs, and constant current. In this work, we present the 1-C CC-CV charging profile in order to validate the model's accuracy. In addition, we test the model with different levels of pulse profiles in the specific SOC region. The root mean square error (RMSE) is used as a metric to quantify the performance of the electrochemical–thermal model simulator summarized



(a)



(b)

Fig. 2. DFN model validation with experimental measurements in terms of temperature and voltage outputs. (a) Experiment setup. (b) Model validation.

TABLE I
DFN MODEL VALIDATION RESULTS

Profile	Voltage RMSE [mV]	Temperature RMSE [°C]
1C CC-CV	34.2	0.9446
1C Pulse	8.5	0.1899
1.5C Pulse	5.2	0.0659
2C Pulse	7.6	0.2054

in Table I. We notice that the simulation exhibits physical cell behavior, and the output measurement errors are in the acceptable ranges, i.e., voltage RMSE $\leq$ 50 mV and temperature RMSE $\leq$ 1 °C according to the existing system identification of Li-ion batteries in the literature [38], [39]. With these validated electrochemical parameters, the learning-based fast-charging protocols are developed in Section V.

V. FAST-CHARGING PROBLEM

The crucial role in the battery management system is to the tradeoff between fast charging and aging while satisfying safety constraints. In this context, the fast charging problem can be described as reaching the final SOC in minimum time without violating constraints. The input current is limited by the hardware configuration of the battery charger, namely,

$$-I_{\max} \leq I(t) \leq 0 \tag{46}$$

with the convention that a negative current is charging the battery. In order to limit the different degradation mechanism of the cell, we consider constraints that have to be satisfied during the whole charging process. First, the cell temperature is not allowed to exceed a maximum temperature, $T_{\max}$, such as

$$T(t) \leq T_{\max} \tag{47}$$

as the temperature is closely related to the SEI layer growth [34]. Furthermore, we aim to avoid lithium plating deposition by constraining the side-reaction overpotential in (42) to be positive. Lithium plating is a particularly harmful phenomenon, which happens when it becomes thermodynamically favorable for lithium to plate onto the surface of the negative electrode particles instead of intercalating [34]. If this phenomenon persists, then dendrites can form, grow, and pierce through the separator, causing a short circuit. Note that this degradation mechanism is aggravated when the battery operates at a low temperature. We impose the side-reaction overpotential constraint as follows:

$$\eta_{\text{sr}}(L^-, t) \geq 0. \tag{48}$$

In consideration of the preceding characteristics, we formulate an optimal control problem as follows:

$$\begin{aligned} \max_{I(t)} \quad & -t_f \\ \text{s.t.} \quad & \text{battery dynamics, } (16)-(25) \\ & \text{input constraint, } (46) \\ & \text{state constraints, } (47)-(48) \\ & V(t_0) = V_0,\ T(t_0) = T_0 \\ & \text{SOC}^-(t_f) = \text{SOC}_{\text{ref}} \end{aligned} \tag{49}$$

where $t_0 = 0$ and t_f are the initial and final times of the charging procedure, V_0 and T_0 are the initial values for voltage and temperature, respectively, and SOC_{ref} is the reference SOC at which the charging is considered to completed. This formulation is also referred to as a minimum-time charging problem. Notice that the problem (49) becomes a large-scale optimal control problem when the full-order electrochemical battery model is considered. This is because the model comprises hundreds of time-derivative states and algebraic states describing the internal behaviors of the Li-ion battery. This fact motivates to explore the ADP approach where the value function is estimated from generated samples while improving the controller performance.

For the case in which only output measurements are available (output feedback policy), we cannot consider the positivity constraint on the side-reaction overpotential since it is an unmeasured state. Therefore, the constraint (48) is

TABLE II
ACTOR–CRITIC HYPER PARAMETERS

Variable	Description	Value
γ	Discount factor	0.99
η_π, η_Q	Learning rate of actor, critic	10^{-4}, 10^{-3}
τ	Soft update of target networks	10^{-3}

replaced with a more conservative constraint that limits the voltage below a predefined threshold $V_{\max}$, as follows:

$$V(t) \leq V_{\max}. \tag{50}$$

VI. RESULTS AND DISCUSSION

In this section, we assess the performance of the RL in simulation when the battery fast charging problem (see Section V) is developed for both state- and output-based configurations. The objective is to compare the proposed actor–critic approach with some benchmark algorithms that have been discussed in the literature and exhibit satisfying results in accomplishing the required task. We consider the modified reference governor (MRG) technique [6] and CC-CT-CV protocol based on a PI controller. As previously stated, we rely on the EM presented in Section III as a battery simulator. The electrochemical parameters of the battery model are obtained from the battery manufacturer in order to represent a real battery cell with a graphite anode and a LiNiMnCoO2(NMC) cathode chemistry. Note that NMC electrochemical parameters used in this work are not disclosed; however, the implementation for proposed RL fast charging can be found by using publicly available electrochemical parameter sets, i.e., graphite anode & LiCoO2 (LCO) cathode in the *code availability* section.

Notice that the structure of the actor–critic networks remains consistent for different charging problems. The actor–critic networks are based on neural network architectures [28] with different numbers of neurons. Specifically, the actor network uses two hidden layers with 20–20 neurons, while the critic network uses two hidden layers with 100–75 neurons. The training hyperparameters are described in Table II.

The reward function is designed according to the optimization problem in (49) with the aim of achieving fast charging while guaranteeing safety

$$r_{t+1} = r_{\text{fast}} + r_{\text{safety}}(\boldsymbol{s}_t, \boldsymbol{a}_t) \tag{51}$$

where r_{fast} is an instantaneous penalty for each time step that passes before the reference SOC is achieved. In addition, a penalty is also introduced at each time step when the safety constraints are violated by means of linear penalty functions [40]. Note that the proposed RL framework does not require specific knowledge of the system, which makes the constraint violation to be experienced by the agent during the training. The model-based RL can be considered to guarantee the robustness for safety-critical applications, such as autonomous driving [41]. In this work, the learning process is allowed to exceed the constraint for learning purposes, and this assumption is valid as the battery is a marginally stable system [42]. As discussed in Section V, the safety constraints enforced in the case of a state-based framework and in the case of an output-based one are different due to the fact that the side-reaction overpotential measurement is not available for the output-based case. Therefore, in the state-based configuration, we have the following safety term in the reward function:

$$r_{\text{safety}}(\boldsymbol{s}_t, \boldsymbol{a}_t) = r_{\eta_{\text{sr}}}(\boldsymbol{s}_t, \boldsymbol{a}_t) + r_{\text{temp}}(\boldsymbol{s}_t, \boldsymbol{a}_t) \tag{52}$$

where

$$r_{\eta_{\text{sr}}}(\boldsymbol{s}_t, \boldsymbol{a}_t) = \begin{cases} \lambda_{\text{sr}}\, \eta_{\text{sr}}(t), & \text{if } \eta_{\text{sr}}(t) < 0 \\ 0, & \text{otherwise} \end{cases} \tag{53}$$

$$r_{\text{temp}}(\boldsymbol{s}_t, \boldsymbol{a}_t) = \begin{cases} \lambda_{\text{temp}}(T(t) - T_{\max}), & \text{if } T(t) \geq T_{\max} \\ 0, & \text{otherwise} \end{cases} \tag{54}$$

where the temperature threshold is set to $T_{\max} = 40$ °C. The fast charging term and penalty function coefficients have been tuned as $r_{\text{fast}} = -0.1$, $\lambda_{\text{sr}} = 10$, and $\lambda_{\text{temp}} = -5$.

For the output-based setting, we substitute the limit on the side-reaction overpotential with the more conservative constraint on the voltage, such as

$$r_{\text{safety}}(\boldsymbol{s}_t, \boldsymbol{a}_t) = r_{\text{volt}}(\boldsymbol{s}_t, \boldsymbol{a}_t) + r_{\text{temp}}(\boldsymbol{s}_t, \boldsymbol{a}_t) \tag{55}$$

where r_{temp} is equivalent to (54), and

$$r_{\text{volt}}(\boldsymbol{s}_t, \boldsymbol{a}_t) = \begin{cases} \lambda_{\text{volt}}(V(t) - V_{\max}), & \text{if } V(t) \geq V_{\max} \\ 0, & \text{otherwise} \end{cases} \tag{56}$$

where $V_{\max} = 4.2$ V as specified in the datasheet, and the coefficient in the voltage constraint is set to $\lambda_{\text{volt}} = -100$.

The current is limited within the range $[-2.5\text{C}, 0]$, where C is the C-rate related to the considered cell. In particular, the current constraint is imposed by considering that the agent's action is inherently bounded within the open interval $(-1, 1)$ due to the choice of its last layer as a hyperbolic tangent operator.

A. State-Based Learning Policy

We first] consider the state-based RL approach. Although deep RL has the distinct advantage of side-stepping Bellman's curse of dimensionality for tabular optimal control, it is also known in the literature that a proper features' selection procedure can significantly increase the performance [43]. Therefore, due to the fact that the EM presented in Section III presents a large number of states after discretization, we transform the state-space into a reduced observation space through PCA [44].

1) Principal Component Analysis: Suppose that we have m time-series data samples for the states, $\boldsymbol{s} \in \mathbb{R}^n$, represented as matrix $\boldsymbol{S} \in \mathbb{R}^{n \times m}$. Consider a so-called "principal component" that can be expressed as

$$P = w^T \boldsymbol{S} \tag{57}$$

where $w \in \mathbb{R}^{n \times 1}$ is a vector of weights and $P \in \mathbb{R}^{1 \times m}$ is an arbitrary principal component. If we consider $\boldsymbol{S}$ as a random

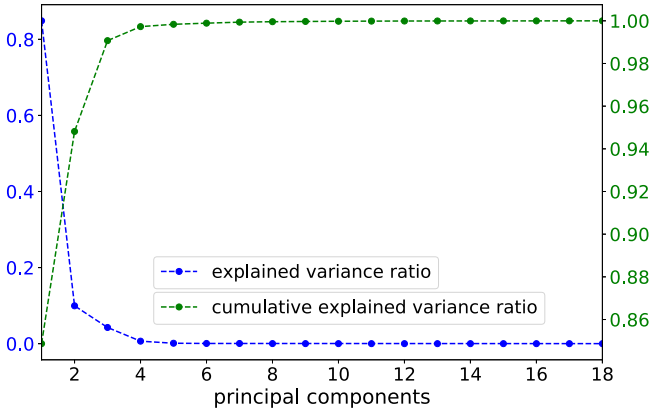


Fig. 3. PCA results on the EM.

matrix, then we seek to choose w to maximize the variance of P

$$\text{var}(P) = w^T \boldsymbol{S}\boldsymbol{S}^T w. \tag{58}$$

We then formulate the following optimization problem while constraining w to have unit length:

$$\begin{aligned} \max_{w} \quad & w^T \boldsymbol{S}\boldsymbol{S}^T w \\ \text{s.t.} \quad & w^T w = 1 \end{aligned} \tag{59}$$

whose solution $w^\star$ yields the first principal component according to $P = (w^\star)^T \boldsymbol{S}$. This method can be extended to compute q principal components, in which the original data $\boldsymbol{s} \in \mathbb{R}^n$ is projected onto a reduced basis of dimension q that maximizes variance [45]. The number of principal components is determined by calculating the ratio of each eigenvalue, Λ_j, to the sum of eigenvalues, $\sum_j \Lambda_j$. The most significant principal component is the one with the largest eigenvalue, which is the most informative. To see how much information we retain through PCA, we calculate the *explained variance ratio* of the principal components as follows:

$$\text{Explained variance ratio} = \frac{\Lambda_j}{\sum_j \Lambda_j}. \tag{60}$$

To apply PCA for dimensionality reduction in the battery model, we first generate a time-series dataset using general charging schemes, i.e., constant current, multistep constant current profiles, in order to collect the evolution of electrochemical states. Then, the state matrix, $\boldsymbol{\mathcal{S}}$, is formulated as the dynamical states in the differential equations (16)–(18). Note that these dynamical states are comprised of hundreds of states after spatial discretization. The goal of PCA is to project the large-scale dynamical system onto a small number of subspaces while maximizing the interpretability of the reduced space. One may wish to use reduced-order EM while preserving specific properties of the high-fidelity model; it is obvious that aging-related constraints, such as lithium plating constraints, may not be preserved in the reduced-order model. Furthermore, the benefit of PCA is to diminish the input size for actor network (policy), which enables to accelerate the training procedure in the deep RL framework. Fig. 3 describes the PCA results on the EM. The general states' evolution of EM can be compressed into five principal components that explain 99.8% of states' information. This can significantly reduce the size of inputs to the actor network by a factor of 100. Based on this dimensionality reduction technique, the learned policy maps the PCA states to input current by interacting with the environment.

2) Training Results of State-Based Policy: In the following, we discuss the performance of the state-based RL approach applied in two different settings of environment temperature, $T_{\text{amb}} = 25$ °C and $T_{\text{amb}} = 15$ °C. Fig. 4 shows the training results of the actor–critic approach, where the shaded region represents the variance of the different variables for each episode over five different initializations of the networks' parameters. The solid lines describe their average value. During the training, an exploration noise is added to the agent's action, and the EM is randomly initialized with $\text{SOC}^-(t_0) \in [0.2,\ 0.4]$ and $T(t_0) \in [25\ °\text{C},\ 32\ °\text{C}]$, thus increasing the exploration capability. Fig. 4(a) depicts the cumulative reward by evaluating the policy every ten episodes without the exploration noise. As it can be noticed, the cumulative reward approaches -3.08 for $T_{\text{amb}} = 25$ °C and -3.51 for $T_{\text{amb}} = 15$ °C. It is important to consider that, although 3000 training episodes are considered, the cumulative reward converges already after 500 cycles. This number of episodes, however, constitutes a large part of the lifetime of a standard Li-ion cell (which is about 1000 cycles). For this reason, it is not feasible to conduct a complete training process directly on a real cell, but, as it is common in RL, the training phase is carried out on a very detailed simulator, and then, only a fine-tuning of the network parameters is done online on the real cell during the first few cycles.

Fig. 4(b) and (c) describes the constraint violations during training. The constraint violation scores are computed according to $\max\{T(t) - T_{\max},\ \forall t \in [0, t_f]\}$ and $\max\{0 - \eta_{\text{sr}}(t),\ \forall t \in [0, t_f]\}$ for each episode. Positive values imply that the constraints are violated during that particular episode. We can see the constraint violation scores approach zero as the episodes increase, which implies that the optimal control policy involves a boundary solution in which these constraints are active (i.e., true with equality during segments of the optimal trajectory). Another interesting interpretation of these figures comes by analyzing which constraint has a greater impact on achieving a fast charging protocol. For instance, the cell temperature constraint is the dominant factor when the ambient temperature is 25 °C, while side-reaction overpotential dominates the charging at $T_{\text{amb}} = 15$ °C. This fact is also reported by the literature stating that low temperatures lead to lithium plating and subsequent lithium dendritic growth [46]. Finally, Fig. 4(d) displays the charging time versus episode number. The charging time decreases to around 15 min at $T_{\text{amb}} = 25$ °C and 17 min at $T_{\text{amb}} = 15$ °C on average. Note that lower ambient temperature takes more time to reach the target SOC as the side reaction overpotential constraint is prone to a violation when the battery is in charging. Note, upon convergence, the proposed actor–critic approach achieves the minimum time goal without violating the safety constraints.

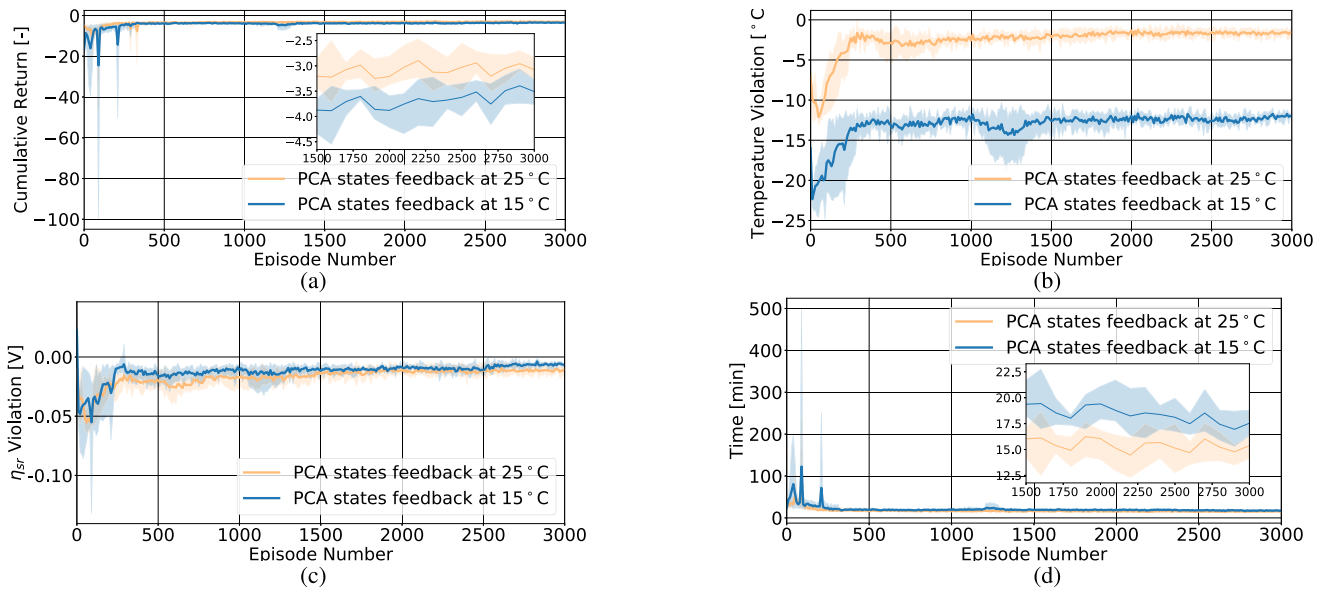


Fig. 4. State-based learning policy results with different initializations of the actor–critic networks at $T_{\text{amb}} = 25$ °C and $T_{\text{amb}} = 15$ °C. (a) Learning curve. (b) Temperature constraint violation. (c) η_{sr}. (d) Average charging time.

It is also informative to examine the charging profiles of SOC, voltage, temperature, and side-reaction overpotential that are obtained by testing the agent after the training process. We consider the initial conditions $\text{SOC}^{-}(t_0) = 0.2$ and $T(t_0) = T_{\text{amb}}$ (for both the cases of $T_{\text{amb}} = 25$ °C and $T_{\text{amb}} = 15$ °C). The results of this analysis are shown in Fig. 5. Moreover, a comparison with an MRG approach [6] is provided. In the MRG approach, the applied current $I(t)$ and the reference current $I^r(t)$ are related according to

$$I(t+1) = \beta(t) I^r(t), \quad \beta \in [0, 1]$$

where $\beta(t)$ is the ratio of reference value that maintains the states in the admissible set. Note that the reference value is equivalent to the allowable charging current in the RL framework, i.e., −2.5 C for comparative analysis. For the 25 °C ambient temperature case, MRG/RL takes 19/20 min to reach the target $\text{SOC}_{\text{ref}} = 0.8$. the For 15 °C ambient temperature case, MRG/RL takes 21.5/22.5 min to reach the 0.8 target $\text{SOC}_{\text{ref}} = 0.8$. The proposed actor–critic approach performs similarly to this MRG benchmark. Note that the MRG assumes perfect knowledge of the model and computes the forward dynamics in the optimization to achieve the goal. In contrast, the state-based actor–critic method is completely model-free. Now, this requires careful qualification. The actor–critic method in this study obviously interacts with a simulation model via iterative episodes. In practice, the actor–critic method could interact with a physical battery, thus avoiding the modeling step. In either case, the results in Fig. 5 assume full state measurement. In a realistic scenario in which only outputs—voltage, temperature, and, in the case of coulomb-counting approximation, also SOC—can be measured, the use of a model-based observer becomes necessary if the state-based RL configuration is adopted. Due to the fact that the development of such a model-based states estimator is a challenging task, we provide in Section VI-C an alternative,

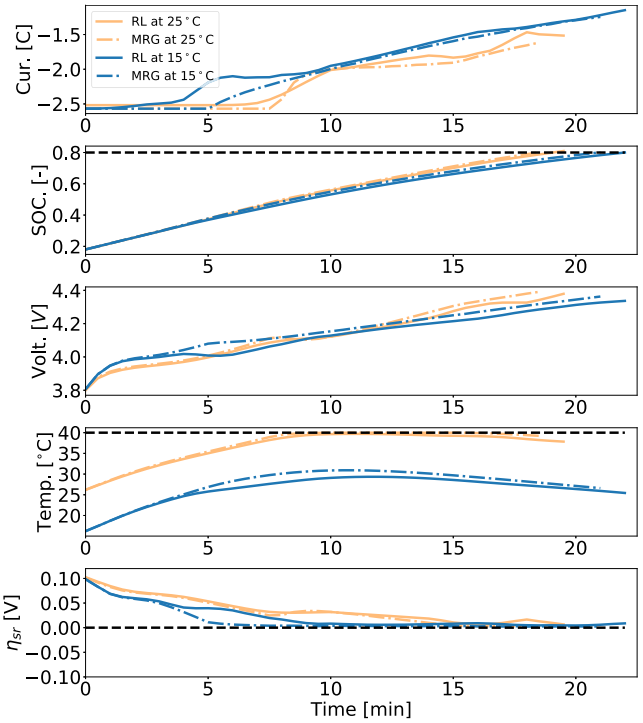


Fig. 5. Validation of state-based learning policy compared with MRG at $T_{\text{amb}} = 25$ °C and $T_{\text{amb}} = 15$ °C.

a completely model-free RL strategy that relies on outputs measurements only.

B. Output-Based Learning Policy

The previously discussed state-based RL charging strategy requires a state estimator to estimate the internal states from the output measurements. However, this is a formidable task as the mathematical structure of the DFN model is formulated

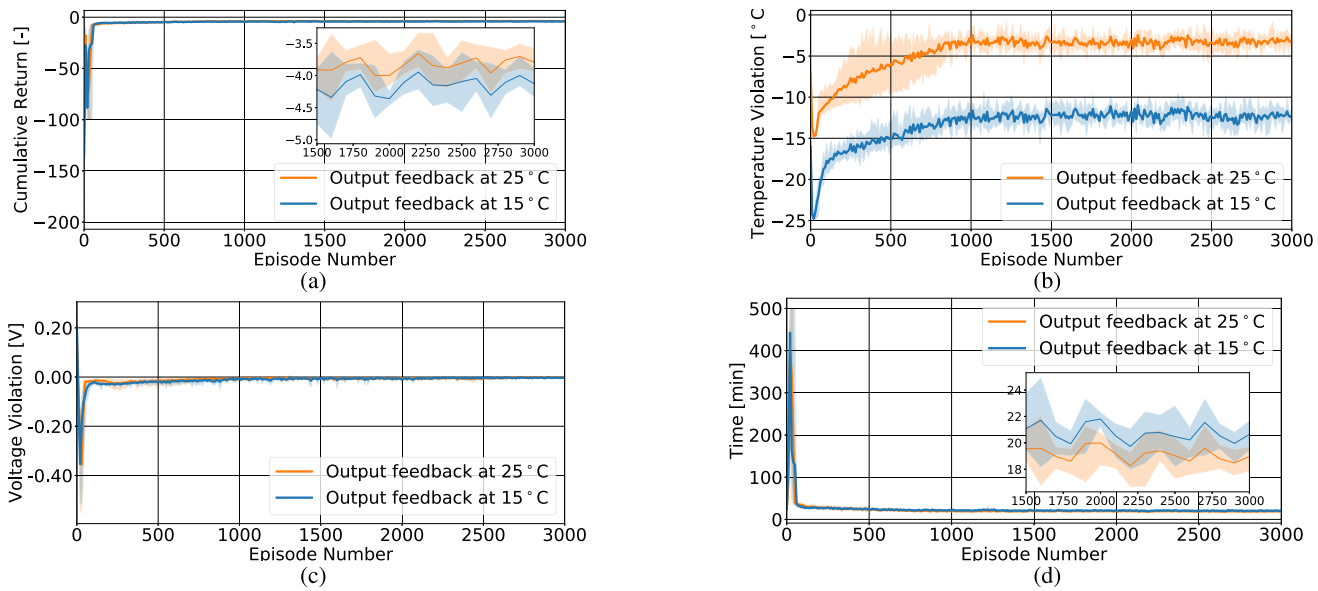


Fig. 6. Output-based learning policy results with different initializations of actor–critic networks at $T_{\text{amb}} = 25$ °C and $T_{\text{amb}} = 15$ °C. (a) Learning curve. (b) Temperature constraint violation. (c) Voltage constraint violation. (d) Average charging time.

as nonlinear DAEs. For this reason, most of the relevant works in the literature on design estimators and controllers are based on reduced-order models. As an alternative, we present, for the first time to the best of our knowledge, a model-free RL charging algorithm based on output measurements only. In particular, in this section, we provide voltage, temperature, and SOC (retrieved through coulomb-counting) as observations to the deep RL agent.

Fig. 6 presents the training results for output-based learning policy using the actor–critic approach at 25 °C and 15 °C ambient temperatures. All the settings are consistent with previous case studies, except for the observations provided to the actor–critic networks, which, here, consists of output measurements only. Also, the safety constraint that enforces a positive side-reaction overpotential is substituted with the more conservative one that limits the terminal voltage below a predefined threshold, as discussed in Section V. In Fig. 6(a), the performances of the learned policy are evaluated every ten episodes with exploration noise removed. The cumulative reward approaches -3.79 and -4.13 for the two ambient temperatures, respectively. Simulation results confirm that the proposed actor–critic approach can be applied for output-based controller design without safety violations. We also notice that the output-based learning policy has fewer neurons in the input layer of the deep neural network compared to state-based policy; however, the level of improvement is not critical since PCA has been applied to the state-based learning policy and decreased the hundreds of states into five.

In particular, Fig. 6(b) and (c) describes the constraint violations during training. The violation scores are calculated according to $\max\{V(t) - V_{\max}, \forall t \in [0, t_f]\}$ and $\max\{T(t) - T_{\max}, \forall t \in [0, t_f]\}$ for each episode. For the output-based policy, the voltage constraint is dominant, at least for the battery model and parameters considered here. The agent is able to learn to ride the voltage constraint while

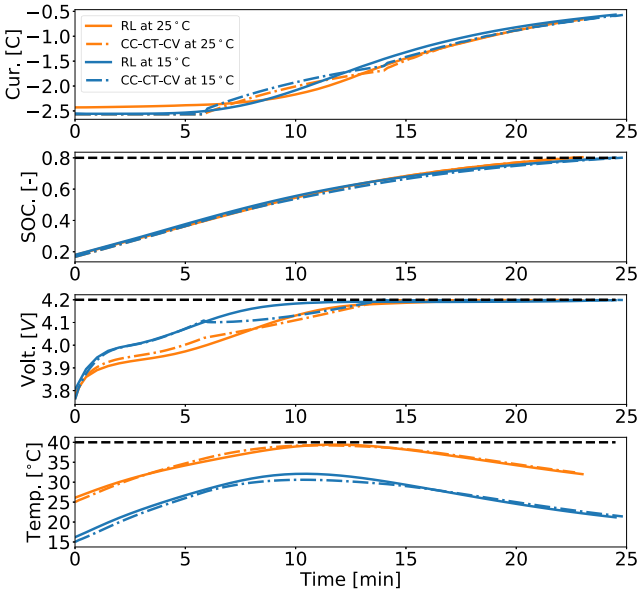


Fig. 7. Validation of the output-based learning policy compared with CC-CT-CV at $T_{\text{amb}} = 25$ °C and $T_{\text{amb}} = 15$ °C.

minimizing the charging time. This is not surprising since it is known that the voltage constraint is particularly conservative. Finally, Fig. 6(d) displays the charging time for achieving the reference SOC. The charging time decreases to around 19 min at $T_{\text{amb}} = 25$ °C and 21 min at $T_{\text{amb}} = 15$ °C ambient temperature on average. We also observe that the charging time takes longer when the ambient temperature is lower, which is consistent with the state-based policy case.

In the following, we focus on the charging profile obtained from the policy at the end of the training process. Moreover, we compare to another output-feedback model-free scheme

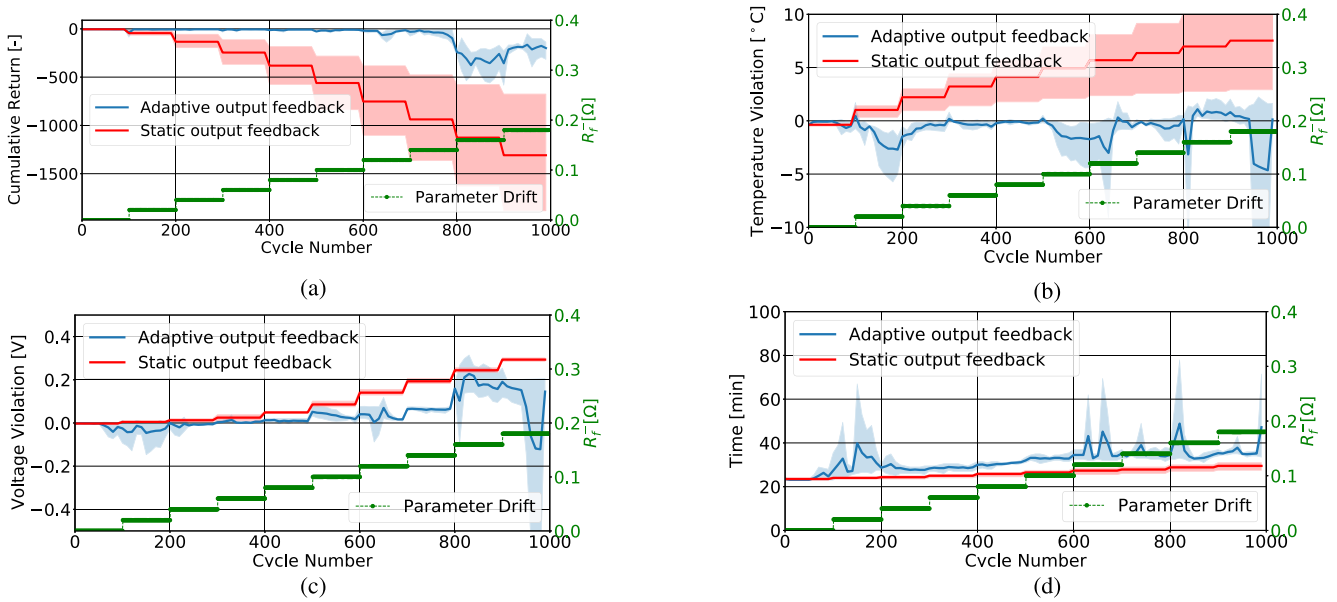


Fig. 8. Adaptive output-based learning policy results in response to the resistance growth due to the aging. The actor–critic networks are obtained from different initializations trained at $T_{\text{amb}} = 25$ °C and $T_{\text{amb}} = 15$ °C. (a) Learning curve. (b) Temperature constraint violation. (c) Voltage constraint violation. (d) Average charging time.

that is based on PI controller as a baseline. This PI controller achieves CC-CT-CV protocol for battery charging. The CC-CT-CV algorithm is defined as

$$I(t) = \begin{cases} I_{\text{ref}}, & t \le t_{\text{pk}} \\ I_{\text{ref}} + K_p e(t) + K_i \sum_{\tau=0}^{t} e(\tau), & t_{\text{pk}} \le t \le t_{\text{cv}} \\ I_{\text{cv}}, & t > t_{\text{cv}} \end{cases}$$

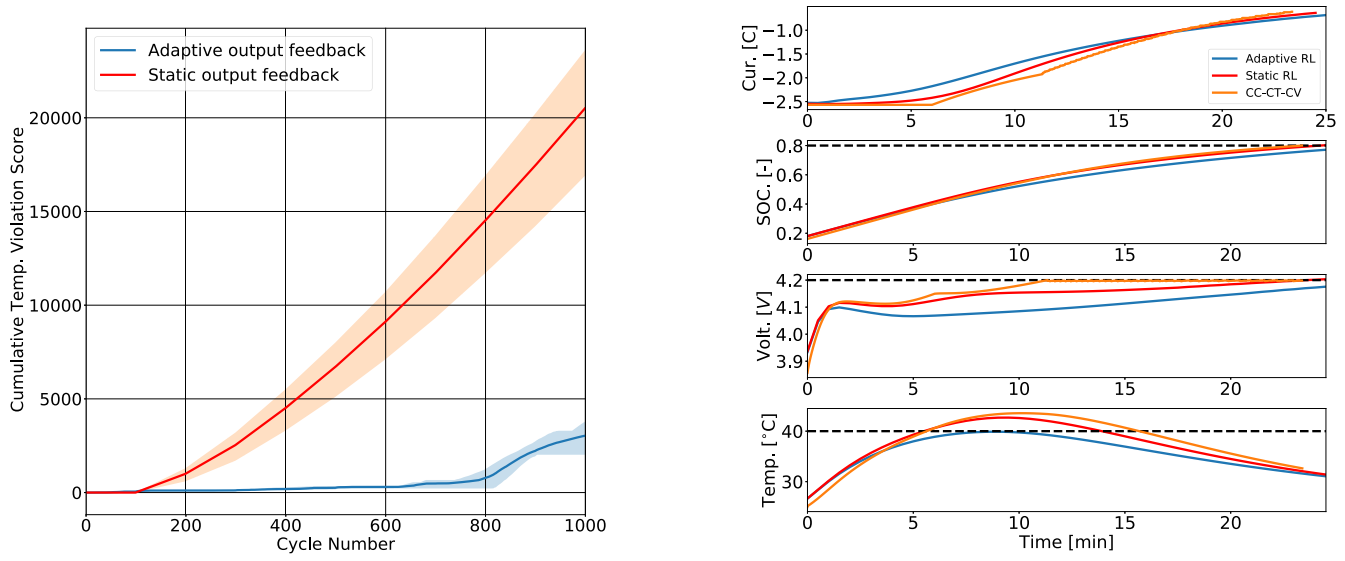
where $e(t)$ is the controller error, i.e., $e(t) = T(t) - T^*$, t_{pk} is the period where constant peak current is applied, i.e., -2.5 C, and t_{cv} is the time when the voltage reaches its maximum. In the CV mode, the current is decreasing in an exponential-like fashion, while the voltage is kept constant until the target SOC is reached. The control gains, K_p and K_i, are obtained by human trial and error. The main defining feature of the CC-CT-CV method is the fact that a temperature constraint is considered, thus enabling a safer charging procedure compared to the conventional CC-CV approach [20]. In this study, we are interested in assessing the RL performance in terms of charging time and state violations compared to the CC-CT-CV method.

We use the same configuration as the previous case study for the initial conditions of the battery. The comparison is shown in Fig. 7. For the 25 °C ambient temperature case, CC-CT-CV/RL takes 23/23.5 min to reach $\text{SOC}_{\text{ref}} = 0.8$, while, for the case of $T_{\text{amb}} = 15$ °C the charging time increases to 25/25 min. We conclude that the output-based actor–critic approach can be used to design an output feedback control for battery fast charging with similar performance to CC-CT-CV. This is due to the fact that the CC-CT-CV profile seems to be the optimal charging procedure in the presence of temperature and voltage constraints. However, the main limitation of the CC-CT-CV relies on the fact that an accurate tuning of the controller is required. Therefore, a possible solution may be the exploitation of RL strategies for the automatic tuning of the PI coefficients in the CC-CT-CV framework. Although we have shown that RL only provided equivalent performance to existing methods when addressing the charging of a fresh battery, we aim to highlight in Section VI-C its ability to adapt the charging control policy in the face of battery aging.

C. Adaptive Output-Based Learning Policy

One of the main challenges facing battery management system development is the controller's adaptability to changes in the cell behavior as it cycles. The proposed deep RL framework exhibits the advantage of learning the optimal policy by directly interacting with the surrounding environment. Specifically, it can adapt to slow changes in the battery parameters due to aging. First, we analyze the adaptability of RL by comparing the fixed parameters of actor–critic called *static* RL policy with an *adaptive* RL policy that continues to adjust the actor–critic network parameters as the cell ages. Note that exploration noise is not considered in this adaptive study because the controller is updated from the estimates of the critic, not an exploration.

First, Fig. 8 describes the results in terms of rewards, state violation, and average charging time in the adaptive test study. To emulate the aging mechanism, we increase the film resistance in the anode, R_f^-, every 100 cycles, up to 1000 cycles. The battery aging mechanisms are a relatively slow process compared to battery dynamics and require a large number of cycles to observe. Furthermore, the degradation process is sophisticated and difficult to identify the root cause of a specific mechanism during cycling. The resistance growth during cycling is a well-known aging phenomenon, but this is one of many observations. The objective of this adaptive test study is to analyze the adaptive behavior of output-based learning policy by perturbing one of the internal



(a) (b)

Fig. 9. Verification and validation of adaptive output-based learning policy. (a) Comparison of cumulative temperature constraint violation score over 1000 cycles. (b) Validation of adaptive output-based learning policy compared with CC-CT-CV at T_{amb}.

electrochemical parameters in the environment, which ultimately affects the voltage and temperature measurements. The actor–critic networks are initialized by the last training results from previous output-based learning policy studies. Both the static and adaptive RL policies achieve the same cumulative return for the first 100 cycles since there are no parameter changes in the environment. However, the static output-based controller fails as the battery is aged, while the adaptive one adapts its control policy through learning, as shown in Fig. 8(a). The drops in the reward can be explained by states' violations in Fig. 8(b) and (c). Note that, after a certain number of cycles, the battery reaches voltage and temperature limits quickly due to aging, which makes it difficult to adapt the learning policy. Nevertheless, the adaptive output feedback policy exhibits better performance than the static (no-updating) output feedback policy. We can also observe that the RL policy is capable of adapting by updating the parameters of actor–critic from the reward (penalty). Fig. 8(d) indicates that the charging time increases from 23.4 to 40.1 min on average at cycle 1000 due to battery aging.

Second, Fig. 9(a) describes what would be the resultant of repeated constraint violations for each cycle throughout battery life. The temperature violation score at each cycle is computed as $\sum\{T(t) - T_{\max} \mid T(t) \geq T_{\max} \forall t \in [0, t_f]\}$, and its cumulative score is plotted in the Fig. 9(a). It is obvious that using an adaptive characteristic of a learning-based controller could mitigate the excessive temperature violation and heat generation. This adaptive controller study can be further extended to design a thermal management system in the future.

Finally, we validate the adaptability of the RL method by comparing it with CC-CT-CV, as shown in Fig. 9(b). From this figure, we can clearly see that adaptive RL can satisfy the constraints, while static RL and CC-CT-CV violate the temperature constraint during the charging process. For the CC-CT-CV method, one might need to tune the PI controller gains heuristically. Consequently, RL (actor–critic) demonstrates a distinct advantage as an adaptive battery fast-charging controller that adjusts as the cell ages.

TABLE III
SUMMARY OF CHARGING TIME [MIN] AND ADAPTABILITY FEATURE

Strategy	Condition	States-based		Outputs-based	
		RL	MRG	RL	CC-CT-CV
Model-based	25°C	20 min	19 min	-	-
	15°C	22.5 min	21.5 min	-	-
Model-free	25°C	-	-	23.5 min	23 min
	15°C	-	-	25 min	25 min
Adaptability	25°C	YES	-	YES	NO

D. Discussion and Remarks

We show that the proposed RL framework can be utilized for solving the battery fast charging problem using both state- and output-based frameworks. The overall results are summarized in Table III. We make several remarks in this section for readers. First, the availability of full state information enables faster charge times than output feedback only since one can consider the side-reaction overpotential constraint instead of the more conservative voltage constraint. This motivates the design of state/parameter estimators as a very important component for reducing the charging time and carefully monitoring immeasurable degradation mechanisms, such as lithium plating. Second, temperature plays a significant role in the battery management system, as it affects both side-reaction overpotential and terminal voltage. The battery thermal management system is another key component for battery fast charging. Third, deep RL, commonly associated with artificial intelligence, can be applied to the battery management system. This work demonstrates that RL is comparable with other existing approaches, and its adaptability to aging is

well-suited for battery fast-charging applications. Finally, the detailed EM that matches with the actual cell can provide an opportunity to investigate the delayed reward problem in the real-world application. The aging-related reward, e.g., capacity retention, can be considered as a sparse reward signal measured by the end of cycles. This is an open research question on how to design a controller that maximizes the delayed reward in the context of the health-aware battery charging problem.

The key advantages of RL are: 1) online adaptation and 2) model-free RL algorithms that are not specialized to a particular battery model/cell design. The methods can be applied to real BMS production in two ways.

The first approach is to use the RL algorithm offline with simulations or physical cells in the lab to determine an optimal charging profile. These RL-optimized profiles can often be represented by a simple protocol, such as CC-CT-CV, or a multiconstant current-current-CV (CC1-CC2-CV) profile. This simple profile can be programmed into an existing BMS system. In other words, an implementation-ready approximation of the optimal result can be obtained via off-line RL.

The second approach is to use RL as *adapt* an existing fast charging protocol as the cell ages. For example, one could use DDPG in the following way. Fit an actor and Q-function to a default fast charging protocol (e.g., CC-CV). Then, as the cell cycles, use DDPG to adapt this policy slowly based on reward signals collected online to improve the performance online.

VII. Conclusion

In this article, we propose a model-free deep RL framework for solving the battery fast-charging problem in the presence of safety constraints when a detailed EM is used as a battery simulator. Among the RL paradigms, the actor–critic scheme and, specifically, the DDPG algorithm have been adopted due to its ability to deal with continuous state and action spaces. To address the state constraints, the reward function has been designed such that the agent learns constraint violations. First, we assume full state measurements and compare a state-based RL algorithm with a reference governor approach considered as a state-of-the-art benchmark. Subsequently, a more realistic scenario in which only output measurements are available is taken into account. In this case, instead of relying on a state estimator, which is challenging to design, we formulate an output-based configuration of the same RL approach. This relies on measurements of voltage, temperature, and SOC (via Coulomb counting). This latter strategy is compared against a CC-CT-CV approach, which can be considered a benchmark state-of-the-art output feedback model-free approach. For both the RL formulations, simulation results show that RL performs similarly to the state-of-the-art benchmarks. Finally, the main advantage of RL is adaptation. Namely, we analyze the output-based RL strategy in the presence of changing battery parameters that mimic battery aging. The results highlight that the proposed approach is able to achieve reasonable performance as the environment changes according to aging throughout the whole battery life, while the CC-CT-CV approach eventually violates safety constraints. Ongoing work involves experimental validation of the proposed framework using a hardware-in-the-loop testing environment with different objectives. To the best of our knowledge, this is the first work that combines artificial intelligence and battery management system. With the open-sourced implementation, one could improve the learning performance using an advanced RL algorithm, i.e., soft actor–critic (SAC) [47], or explore different perspectives of RL, such as inverse RL [48] and hierarchical RL [49].

Code Availability

The code for the proposed deep RL framework for fast charging is publicly available at https://github.com/saehong/RL-BATT-DDPG.

Acknowledgment

The authors thank the LG Chem researchers for their support and discussion in the work.

References

[1] L. Lu, X. Han, J. Li, J. Hua, and M. Ouyang, "A review on the key issues for lithium-ion battery management in electric vehicles," *J. Power Sources*, vol. 226, pp. 272–288, Mar. 2013.

[2] R. C. Cope and Y. Podrazhansky, "The art of battery charging," in *Proc. 14th Annu. Battery Conf. Appl. Adv.*, Jan. 1999, pp. 233–235.

[3] H. He, R. Xiong, and J. Fan, "Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach," *Energies*, vol. 4, no. 4, pp. 582–598, Mar. 2011.

[4] S. Santhanagopalan, Q. Guo, P. Ramadass, and R. E. White, "Review of models for predicting the cycling performance of lithium ion batteries," *J. Power Sources*, vol. 156, no. 2, pp. 620–628, 2006.

[5] R. Klein, N. A. Chaturvedi, J. Christensen, J. Ahmed, R. Findeisen, and A. Kojic, "Optimal charging strategies in lithium-ion battery," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2011, pp. 382–387.

[6] H. Perez, N. Shahmohammadhamedani, and S. Moura, "Enhanced performance of Li-ion batteries via modified reference governors and electrochemical models," *IEEE/ASME Trans. Mechtron.*, vol. 20, no. 4, pp. 1511–1520, Aug. 2015.

[7] M. Torchio *et al.*, "Real-time model predictive control for the optimal charging of a lithium-ion battery," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2015, pp. 4536–4541.

[8] C. Zou, X. Hu, Z. Wei, T. Wik, and B. Egardt, "Electrochemical estimation and control for lithium-ion battery health-aware fast charging," *IEEE Trans. Ind. Electron.*, vol. 65, no. 8, pp. 6635–6645, Aug. 2018.

[9] H. E. Perez, X. Hu, S. Dey, and S. J. Moura, "Optimal charging of Li-ion batteries with coupled electro-thermal-aging dynamics," *IEEE Trans. Veh. Technol.*, vol. 66, no. 9, pp. 7761–7770, Sep. 2017.

[10] A. Pozzi, M. Torchio, and D. M. Raimondo, "Film growth minimization in a Li-ion cell: A pseudo two dimensional model-based optimal charging approach," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2018, pp. 1753–1758.

[11] X. Lin, S. Wang, and Y. Kim, "A framework for charging strategy optimization using a physics-based battery model," *J. Appl. Electrochem.*, vol. 49, no. 8, pp. 779–793, Aug. 2019.

[12] A. Pozzi, G. Ciaramella, S. Volkwein, and D. M. Raimondo, "Optimal design of experiments for a lithium-ion cell: Parameters identification of an isothermal single particle model with electrolyte dynamics," *Ind. Eng. Chem. Res.*, vol. 58, no. 3, pp. 1286–1299, Jan. 2019.

[13] S. Park, D. Kato, Z. Gima, R. Klein, and S. Moura, "Optimal experimental design for parameterization of an electrochemical lithium-ion battery model," *J. Electrochem. Soc.*, vol. 165, no. 7, pp. A1309–A1323, 2018.

[14] W. Waag, C. Fleischer, and D. U. Sauer, "Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles," *J. Power Sources*, vol. 258, pp. 321–339, Jul. 2014.

[15] P. H. L. Notten, J. H. G. O. H. Veld, and J. R. G. van Beek, "Boostcharging Li-ion batteries: A challenging new charging concept," *J. Power Sources*, vol. 145, no. 1, pp. 89–94, 2005.

[16] S. S. Zhang, K. Xu, and T. R. Jow, "Study of the charging process of a $LiCoO_2$-based Li-ion battery," *J. Power Sources*, vol. 160, no. 2, pp. 1349–1354, Oct. 2006.

[17] C. H. Lin, C. Y. Hsieh, and K. H. Chen, "A Li-ion battery charger with smooth control circuit and built-in resistance compensator for achieving stable and fast charging," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 2, pp. 506–517, Feb. 2010.

[18] Y. Yin, Y. Hu, S.-Y. Choe, H. Cho, and W. T. Joe, "New fast charging method of lithium-ion batteries based on a reduced order electrochemical model considering side reaction," *J. Power Sources*, vol. 423, pp. 367–379, May 2019.

[19] P. M. Attia *et al.*, "Closed-loop optimization of fast-charging protocols for batteries with machine learning," *Nature*, vol. 578, no. 7795, pp. 397–402, Feb. 2020.

[20] L. Patnaik, A. V. J. S. Praneeth, and S. S. Williamson, "A closed-loop constant-temperature constant-voltage charging technique to reduce charge time of lithium-ion batteries," *IEEE Trans. Ind. Electron.*, vol. 66, no. 2, pp. 1059–1067, Feb. 2019.

[21] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[22] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[23] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1889–1897.

[24] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. 30th AAAI Conf. Artif. Intell.*, Mar. 2016, pp. 2094–2100.

[25] F. Belletti, D. Haziza, G. Gomes, and A. M. Bayen, "Expert level control of ramp metering based on multi-task deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1198–1207, Apr. 2018.

[26] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Syst.*, vol. 32, no. 6, pp. 76–105, Dec. 2012.

[27] S. Park *et al.*, "Reinforcement learning-based fast charging control strategy for Li-ion batteries," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2020, pp. 100–107.

[28] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.

[29] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 1057–1063.

[30] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1. Belmont, MA, USA: Athena Scientific, 2005, no. 3.

[31] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 387–395.

[32] K. E. Thomas, J. Newman, and R. M. Darling, "Mathematical modeling of lithium batteries," in *Advances in Lithium-Ion Batteries*. Boston, MA, USA: Springer, 2002, pp. 345–392.

[33] N. A. Chaturvedi, R. Klein, J. Christensen, J. Ahmed, and A. Kojic, "Algorithms for advanced battery-management systems," *IEEE Control Syst. Mag.*, vol. 30, no. 3, pp. 49–68, Jun. 2010.

[34] A. Tomaszewska *et al.*, "Lithium-ion battery fast charging: A review," *eTransportation*, vol. 1, Aug. 2019, Art. no. 100011.

[35] P. Arora, M. Doyle, and R. E. White, "Mathematical modeling of the lithium deposition overcharge reaction in lithium-ion batteries using carbon-based negative electrodes," *J. Electrochem. Soc.*, vol. 146, no. 10, p. 3543, 1999.

[36] J. C. Forman, S. Bashash, J. L. Stein, and H. K. Fathy, "Reduction of an electrochemistry-based Li-ion battery model via quasi-linearization and Padé approximation," *J. Electrochem. Soc.*, vol. 158, no. 2, pp. A93–A101, 2011.

[37] G. Fan, K. Pan, and M. Canova, "A comparison of model order reduction techniques for electrochemical characterization of lithium-ion batteries," in *Proc. 54th IEEE Conf. Decis. Control (CDC)*, Dec. 2015, pp. 3922–3931.

[38] J. C. Forman, S. J. Moura, J. L. Stein, and H. K. Fathy, "Genetic identification and Fisher identifiability analysis of the Doyle–Fuller–Newman model from experimental cycling of a $LiFePO_4$ cell," *J. Power Sources*, vol. 210, pp. 263–275, Jul. 2012.

[39] S. B. Lee and S. Onori, "A robust and sleek electrochemical battery model implementation: A MATLAB® framework," *J. Electrochem. Soc.*, vol. 168, no. 9, Sep. 2021, Art. no. 090527.

[40] A. E. Smith, D. W. Coit, T. Baeck, D. Fogel, and Z. Michalewicz, "Penalty functions," *Handbook Evol. Comput.*, vol. 97, no. 1, p. C5, 1995.

[41] F. Berkenkamp, M. Turchetta, A. P. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees," 2017, *arXiv:1705.08551*.

[42] S. J. Moura, "Estimation and control of battery electrochemistry models: A tutorial," in *Proc. 54th IEEE Conf. Decis. Control (CDC)*, Dec. 2015, pp. 3906–3912.

[43] M. A. Hall and L. A. Smith, "Practical feature subset selection for machine learning," in *Proc. 21st Australas. Comput. Sci. Conf.*, 1998, pp. 181–191.

[44] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graph. Statist.*, vol. 15, no. 2, pp. 265–286, 2004.

[45] G. C. Calafiore and L. El Ghaoui, *Optimization Models*. Cambridge, U.K.: Cambridge Univ. Press, 2014.

[46] T. Waldmann, M. Wilka, M. Kasper, M. Fleischhammer, and M. Wohlfahrt-Mehrens, "Temperature dependent ageing mechanisms in lithium-ion batteries—A post-mortem study," *J. Power Sources*, vol. 262, pp. 129–135, Sep. 2014.

[47] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.

[48] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan, "Cooperative inverse reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 3909–3917.

[49] A. S. Vezhnevets *et al.*, "Feudal networks for hierarchical reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3540–3549.