Reinforcement Learning-based Fast Charging Control Strategy for Li-ion Batteries

Saehong Park, Andrea Pozzi, Michael Whitmeyer, Hector Perez,
Won Tae Joe, Davide M Raimondo, Scott Moura

***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 can fall into solving large-scale nonlinear optimal control problems according to a battery model. Within this context, several model-based techniques have been proposed in the literature. However, the effectiveness of such strategies is significantly limited by model complexity and uncertainty. Additionally, it is difficult to track parameters related to aging and re-tune the model-based control policy. With the aim of overcoming these limitations, in this paper we propose a fast-charging strategy subject to safety constraints which relies on a model-free reinforcement learning framework. In particular, we focus on the policy gradient-based actor-critic algorithm, i.e., deep deterministic policy gradient (DDPG), in order to deal with continuous sets of actions and sets. The validity of the proposal is assessed in simulation when a reduced electrochemical model is considered as the real plant. Finally, the online adaptability of the proposed strategy in response to variations of the environment parameters is highlighted with consideration of state reduction.**

***Keywords*—Reinforcement learning, Actor-critic, Electrochemical model, Battery charging, Optimal control,**

I. Introduction

Lithium-ion batteries are crucial technologies for electrified transportation, clean power systems, and consumer electronics. Although among all the different chemistries, Li-ion batteries exhibit promising features in terms of energy and power density, they still present limited capacity and long charging time. While the former is mostly related to the battery chemistry and design phase, the latter depends on the employed charging strategy. Within this context, the trade-off 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. For this reason, several model-based optimal control techniques have been proposed in the literature with the aim of providing fast-charging while guaranteeing safety constraints.

Saehong Park, Michael Whitmeyer, Hector Perez, and Scott Moura are with the Energy, Controls and Applications Lab (eCAL) at the University of California, Berkeley, CA 94720, USA (E-mail: {sspark,mwhitmeyer,heperez,smoura}@berkeley.edu)

Andrea Pozzi and Davide M Raimondo is with University of Pavia, Corso Str. Nuova, 65, 27100, Pavia, Italy (E-mail: andrea.pozzi03@universitadipavia.it, davide.raaimondo@unipv.it)

Won Tae Joe is with LG Chem, BMS Advanced SW Project Team, Yuseong-gu, Daejeon, 305-738, South Korea (E-mail: wontaejoe@lgchem.com)

The authors in [1] formulate a minimum-time charging problem and use nonlinear model predictive control. Similarly, authors in [2] propose quadratic dynamic matrix control formulation to design an optimal charging strategy for real-time model predictive control. In the context of aging mechanism, the authors of [3] have studied the trade-off between charging speed and degradation, based on an electro-thermal-aging model. The authors in [4] consider minimizing film layer growth of the electrochemical model. Authors in [5], [6] derive an optimal current profile using a single particle model with intercalation-induced stress generation. The key novelty here is incorporating mechanical fracture, which can be a dominant mechanism in degradation and capacity fade. To ensure safety, a proportional–integral–derivative controller is proposed. On the other hand, the authors in [7] synthesize a state estimation and model predictive control scheme for a reduced electrochemical-thermal model, in order to design 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 [8], the authors exploit differential flatness properties of the single particle model to yield a computationally efficient optimal control problem, solved via pseudospectral methods.

However, the exploitation of model-based charging procedure has to face some crucial challenges. (i) Every model is inherently subject to modeling mismatches and uncertainties. (ii) The most commonly used detailed models for Li-ion batteries are the electrochemical ones which typically contain hundreds or thousands of states, leading to a large-scale optimization problem. (iii) The model parameters drift as the battery ages. It is important to notice that most of the model-based strategies proposed in the literature rely on simplified electrochemical models (the few ones which implement full order models represent the boundary of what can be done in this area) and almost none of them consider adaptability of the control strategy to variations in the parameters. In addition, electrochemical models present observability and identifiability issues [9], which often lead to the necessity of optimally designing the experiments which have to be conducted in order to properly estimate the parameters with a sufficiently high accuracy [10], [11]. All these issues can be addressed by using a charging procedure based on a model-free Reinforcement Learning (RL) framework [12]. 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 (a.k.a. the state). The model-free property implies that the agent learns online the feedback control policy, directly from

interactions with the environment, such as reward and state observation. Such policy is iteratively updated in order to maximize the expected long-term reward. Notice that, the reward has to be properly designed in order to make the agent learn how to accomplish the required task.

Most RL algorithms can be classified in two different groups: tabular methods, e.g., Q-learning, SARSA, and approximate solutions methods which is also called "Approximate Dynamic Programming (ADP)". While the former performs well only in presence of small and discrete set of actions and states, the latter can be used even with continuous state and action spaces solving the so-called "curse of dimensionality". On the other hand, the convergence of the former is proven under mild assumptions. However, no proof of convergence exists for the approximate methods in the general case. The recent success in several applications of RL based on deep neural networks as function approximators has greatly increased expectations in the scientific community [13]–[16]. From a control systems perspective, the design of RL algorithms involves feedback control laws for dynamical systems via optimal adaptive control methods [17]. It is also important to notice that several works have been focused in developing safe-RL strategies, which are able to learn optimal control policy while guaranteeing safety constraints [18], which are fundamental in the context of battery fast-charging.

In this paper, a fast-charging strategy subject to safety constraints, using a model-free reinforcement learning framework, is proposed for the first time to the knowledge of the authors in the context of Li-ion batteries. The use of such a methodology enables adaptation to uncertain and drifting parameters. Moreover, the exploitation of ADP-based approaches allows one to mitigate the curse of dimensionality for large-scale nonlinear optimal control problems by adopting parameterized actor/critic networks. In particular, we focus on the Deep Deterministic Policy Gradient (DDPG) [19] algorithm, which is an actor-critic formulation suitable for the case of continuous actions space and includes deep neural networks as function approximators. The safety constraints are considered by including a penalty in the reward function in case of violation. The control technique is tested by considering a Single Particle Model with Electrolyte and Thermal (SPMeT) [20] dynamics as the battery simulator. Two different scenarios are presented: in the first one all the states are assumed measurable from the agent, while in the second this assumption is dropped and only state of charge and temperature are considered available. The results show that the RL-agent is able to achieve high performance in both the scenarios. Finally, we examine the online adaptability of the proposed methodology in the case of varying parameters, i.e. degradation.

The paper is organized as follows. Section II briefly presents the reinforcement learning approach. Section III describes the battery models and control problem formulation. Section IV presents case study with simulation results. In Section V, we summarize our work and provide perspectives on future work.

II. Reinforcement Learning Approach

In this section a standard reinforcement learning setup is presented, along with the main feature of Approximate Dynamic Programming and actor-critic algorithm.

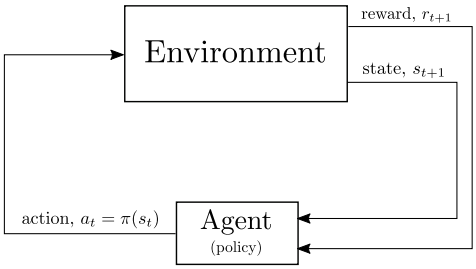

Figure 1: Reinforcement Learning framework.

A. Markov Decision Process, Policy and Value Functions

In the reinforcement learning framework shown in Fig. 1, we seek the best policy that will maximize the total rewards received from the environment E (i.e. plant). At each time step $t \in \mathbb{R}^+$ the environment exhibits state vector, $\boldsymbol{s}_t \in \mathcal{S}$, where $\mathcal{S}$ is the state space, the control policy (a.k.a. agent) observes the states $\boldsymbol{s}_t$ and picks an action $\boldsymbol{a}_t \in \mathcal{A}$, with $\mathcal{A}$ being the action space. This action is executed on 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 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:

$$V^\pi(\boldsymbol{s}_t) \doteq \mathbb{E}_{r_{i>t},\, s_{i>t} \sim E,\, a_{i\geq t} \sim \pi} \Big[R_t \mid \boldsymbol{s}_t \Big] \tag{2}$$

Then, the optimal policy π^* is the policy that corresponds to the maximum value of the value function

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

The solution of (3) is pursued by those methods which follow the Dynamic Programming (DP) paradigm. Such paradigm assumes a perfect knowledge of the environment E (i.e, the state-transition probability as well as the reward function are known).

The next definition, known as the "Q-function," plays a crucial role in model-free reinforcement learning. Consider the *state-action value function*, $Q^\pi(\boldsymbol{s}_t, \boldsymbol{a}_t)$, which is a function of the state-action pair and returns a real value. This Q-value corresponds to the long-term expected return when action $\boldsymbol{a}_t$ is taken in state $\boldsymbol{s}_t$, and then the policy π is followed henceforth. Mathematically,

$$Q^\pi(\boldsymbol{s}_t, \boldsymbol{a}_t) \doteq \mathbb{E}_{r_{i>t},\, s_{i>t} \sim E,\, a_{i>t} \sim \pi} \Big[R_t \mid \boldsymbol{s}_t,\, \boldsymbol{a}_t \Big] \tag{4}$$

The state-action value function can be expressed as Bellman equation, such as:

$$Q^{\pi}(\boldsymbol{s}_t, \boldsymbol{a}_t) = \mathbb{E}_{r_{i>t},\, s_{i>t} \sim E}\Big[r(\boldsymbol{s}_t, \boldsymbol{a}_t) + \gamma \mathbb{E}_{a_{t+1}\sim\pi}\big[Q^{\pi}(\boldsymbol{s}_{t+1}, \boldsymbol{a}_{t+1})\big]\Big] \tag{5}$$

The optimal Q-function $Q^*(\boldsymbol{s}_t, \boldsymbol{a}_t)$ gives the expected total reward received by an agent that starts in $\boldsymbol{s}$, picks (possibly non-optimal) action $\boldsymbol{a}_t$, and then behaves optimally afterwards. $Q^*(\boldsymbol{s}_t, \boldsymbol{a}_t)$ indicates how good it is for an agent to pick action $\boldsymbol{a}_t$ while being in state $\boldsymbol{s}_t$. Since $V^*(\boldsymbol{s}_t)$ is the maximum expected total 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. Tabular Methods and Approximated Solutions

In a model-free framework, the Q-function can be learned directly from the interaction with the environment, by means of the reward collected over time. Within this context two different approaches can be considered: tabular methods and ADP [12], [21]. The former store the Q-function as a table whose entrance are the states and the actions, while the latter uses parameterized Q-function using Value Function Approximation (VFA). The main advantage of ADP relies in its ability of solving the so-called curse of dimensionality, which is a negative feature of both DP and reinforcement learning strategies based on tabular methods [22]. In particular, the curse of dimensionality consists on the exponential rise in the time and space required to compute a solution to an MDP problem as the dimension (i.e. the number of state and control variables) increase [23]. Due to such issue the use of both DP and tabular methods is limited to the context of small and discrete action and state spaces.

Let consider the following approximation

$$Q^{\pi}(\boldsymbol{s}_t, \boldsymbol{a}_t) \approx Q(\boldsymbol{s}_t, \boldsymbol{a}_t | \theta^{Q^{\pi}}) \tag{8}$$

The idea used by ADP methods to solve the curse of dimensionality is to seek for the optimal parameters vector $\theta^{Q^\star}$ instead directly for the Q-function $Q^\star(\boldsymbol{s}_t, \boldsymbol{a}_t)$, thus reducing significantly the size of the optimization problem. Several function approximators can be employed, e.g. linear approximators, neural networks, kernel-based functions. One of the most famous example of ADP using deep neural networks as VFAs is given in [13], where the deep Q-learning algorithm is proposed.

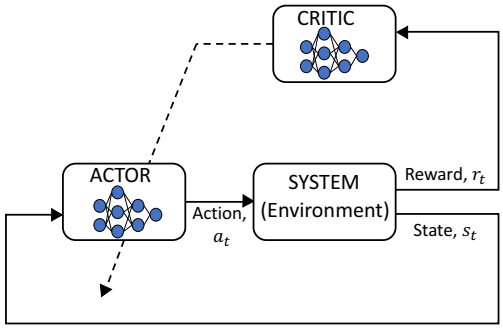

Figure 2: Actor-Critic Structure.

C. Actor-Critic

In RL, the action is taken by a *policy* to maximize the total accumulated reward. By following a given policy and processing the rewards, one should estimate the expected return given states from a *value function*. In the actor-critic approach, the *actor* improves the policy based on the value function that is estimated by the *critic* as depicted in Fig. 2. We specifically focus on the policy gradient-based actor-critic algorithm in this work, and, in particular, on the deep deterministic policy gradient (DDPG) [19]. This algorithm is an extension of deep Q-network (DQN) [13] to continuous actions, maintaining the importance of features such as: (i) random sampling from replay buffer where tuples are saved, (ii) the presence of target networks for stabilizing the learning process. The algorithm begins with a parameterized critic network, $Q(\boldsymbol{s}_t, \boldsymbol{a}_t | \boldsymbol{\theta}^Q)$, and actor network, $\pi(\boldsymbol{s}_t | \boldsymbol{\theta}^{\pi})$.

1) Critic: The role of the critic is to evaluate the current policy prescribed by the actor. Action is taken from the actor network with exploration noise, namely

$$\boldsymbol{a}_t = \pi(\boldsymbol{s}_t | \boldsymbol{\theta}^{\pi}) + \mathcal{N}_t \tag{9}$$

where $\pi(\boldsymbol{s}_t)$ is a neural network, and $\mathcal{N}_t$ is exploration noise. After applying an action, we observe reward $\boldsymbol{r}_{t+1}$ and next state $\boldsymbol{s}_{t+1}$. The tuple $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_{t+1}, \boldsymbol{s}_{t+1})$ is stored in the replay buffer. We sample a random mini-batch of N transitions from the buffer and for $i = 1, \cdots, N$ we set

$$y_i = r_{i+1} + \gamma Q'(\boldsymbol{s}_{i+1}, \pi'(\boldsymbol{s}_{i+1} | \boldsymbol{\theta}^{\pi'}) | \boldsymbol{\theta}^{Q'}) \tag{10}$$

where superscript $'$ denotes the target network, whose parameters are slowly updated in order to track and filter the ones of the actual network thus reducing the chattering due to the learning process and enhancing its convergence. The critic network is updated to minimize the loss, $\mathcal{L}$:

$$\mathcal{L} = \frac{1}{N} \sum_i \big(y_i - Q(\boldsymbol{s}_i, a_i | \boldsymbol{\theta}^Q)\big)^2 \tag{11}$$

$$\boldsymbol{\theta}_{k+1}^Q = \boldsymbol{\theta}_k^Q + \eta_Q \nabla_{\boldsymbol{\theta}^Q} \mathcal{L} \tag{12}$$

where index-k denotes the gradient descent algorithm iterates, and η_Q denotes the learning rates of the critic network.

2) Actor: The parameters of the actor network are updated in order to maximize the long-term expected reward $\mathcal{J}(\theta^{\pi}) = V^{\pi}(\boldsymbol{s}_t)$ over episodes

$$\boldsymbol{\theta}_{k+1}^{\pi} = \boldsymbol{\theta}_k^{\pi} - \eta_{\pi} \nabla_{\boldsymbol{\theta}^{\pi}} \mathcal{J} \tag{13}$$

where index-k denotes the gradient descent algorithm iterates, and η_{π} denotes the learning rates of the actor network.

Notice that according to the proof in [24], the policy gradient in (13) can be expressed as

$$\nabla_{\boldsymbol{\theta}^\pi}\mathcal{J} \approx \mathbb{E}_{\boldsymbol{s}_t\sim E}\left[\nabla_{\boldsymbol{a}}Q(\boldsymbol{s},\boldsymbol{a}|\boldsymbol{\theta}^Q)|_{\boldsymbol{s}_t,\,\pi(s_t)}\nabla_{\boldsymbol{\theta}^\pi}\pi(\boldsymbol{s}|\boldsymbol{\theta}^\pi)|_{\boldsymbol{s}_t}\right] \tag{14}$$

which is then approximated by samples as follows

$$\nabla_{\boldsymbol{\theta}^\pi}\mathcal{J} \approx \frac{1}{N}\sum_i \nabla_{\boldsymbol{a}}Q(\boldsymbol{s},\boldsymbol{a}|\boldsymbol{\theta}^Q)|_{\boldsymbol{s}_i,\,\pi(s_i)}\nabla_{\boldsymbol{\theta}^\pi}\pi(\boldsymbol{s}|\boldsymbol{\theta}^\pi)|_{\boldsymbol{s}_i} \tag{15}$$

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

$$\begin{aligned}\boldsymbol{\theta}^{Q'} &\leftarrow \tau\boldsymbol{\theta}^Q + (1-\tau)\boldsymbol{\theta}^{Q'}\\ \boldsymbol{\theta}^{\pi'} &\leftarrow \tau\boldsymbol{\theta}^\pi + (1-\tau)\boldsymbol{\theta}^{\pi'}\end{aligned} \tag{16}$$

where τ is the level of "soft-update". Equation (16) improves the stability of the learning procedure. Note that 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.

III. Battery Charging problem

In this section, we briefly discuss the battery models and control problem formulation used for the RL framework. We consider reduced order of electrochemical model that contains a large number of states, but achieves high-accuracy and represents physical details of battery dynamics. We also introduce the battery charging control problem formulation in this section.

A. Electrochemical Model

The Single Particle Model with Electrolyte and Thermal Dynamics (SPMeT) is derived from the Doyle-Fuller-Newman (DFN) electrochemical battery model. The DFN model employs a continuum of particles in both the anode and cathode of the cell. The SPMeT uses a simplified representation of solid phase diffusion that employs a single spherical particle in each electrode. The governing equations for SPMeT include linear and quasiliniar partial differential equations (PDEs) and a strongly nonlinear voltage output equation, given by:

$$\frac{\partial c_s^\pm}{\partial t}(r,t) = \frac{1}{r^2}\frac{\partial}{\partial r}\left[D_s^\pm r^2\frac{\partial c_s^\pm}{\partial r}(r,t)\right], \tag{17}$$

$$\varepsilon_e^j\frac{\partial c_e^j}{\partial t}(x,t) = \frac{\partial}{\partial x}\left[D_e^{\text{eff}}(c_e^j)\frac{\partial c_e^j}{\partial x}(x,t) + \frac{1-t_c^0}{F}i_e^j(x,t)\right], \tag{18}$$

where $t \in \mathbb{R}_+$ represents time. The state variables are lithium concentration in the active particles of both electrode denoted by $c_s^\pm(r,t)$ and lithium concentration in the elctrolyte denoted by $c_e(x,t)$. $D_s^\pm$ and $D_e^{eff}(\cdot)$ are diffusion coefficients for solid phase and liquid phase dynamics. Note that superscript j denotes anode, seperator and cathode, $j \in \{+,\text{sep},-\}$. Input current $I(t)$ is applied to the boundary conditions of governing PDEs. The terminal voltage output is governed by a combination of electric overpotential, electrode thermodyanmics, and Butler-Volmer kinetics, yielding:

$$\begin{aligned}V_{\text{T}}(t) =& \frac{RT_{cell}(t)}{\alpha F}\sinh^{-1}\left(\frac{-I(t)}{2a^+AL^+\bar{i}_0^+(t)}\right)\\ &- \frac{RT_{cell}(t)}{\alpha F}\sinh^{-1}\left(\frac{I(t)}{2a^-AL^-\bar{i}_0^-(t)}\right)\\ &+ U^+\left(c_{ss}^+(t)\right) - U^-\left(c_{ss}^-(t)\right)\\ &- \left(\frac{R_f^+}{a^+AL^+} + \frac{R_f^-}{a^-AL^-}\right)I(t)\\ &- \left(\frac{L^+ + 2L^{sep} + L^-}{2A\bar{\kappa}^{eff}}\right)I(t)\\ &+ k_{conc}(t)[ln(c_e(0^+,t)) - ln(c_e(0^-,t))],\end{aligned} \tag{19}$$

where c_{ss} is the solid phase surface concentration, namely $c_{ss}^\pm(x,t) = c_s^\pm(x,R_s^\pm,t)$, $U^\pm$ is the open-circuit potential, and $c_{s,\max}^\pm$ is the maximum possible concentration in the solid phase. The nonlinear temperature dynamics are modeled with a simple heat transfer equation given by:

$$\frac{dT_{\text{cell}}}{dt}(t) = \frac{\dot{Q}(t)}{mc_{p;th}} - \frac{T_{\text{cell}}(t) - T_\infty}{mc_{p,th}R_{th}} \tag{20}$$

where T_{cell} represents cell temperature, T_∞ is the ambient temperature, m is the mass of the cell, $c_{p,th}$ is the thermal specific heat capacity of the cell, R_{th} is the thermal resistance of the cell, and $\dot{Q}(t)$ is the heat added from the charging, which is governed by

$$\dot{Q}(t) = I(t)((U^+(SOC_p) - U^-(SOC_n)) - V_{\text{T}}(t)) \tag{21}$$

with the convention that a negative current is charging current, and $V(t)$ is the voltage determined by (19). Both nonlinear open circuit potential functions in (21) are functions of the bulk SOC in the anode and cathode, respectively. This heat generation term makes the temperature dynamics nonlinear. In this work, we focus on the SOC in anode expressed as a normalized volume sum along the radial axis:

$$SOC_n = \frac{3}{\bar{c_{s,max}}(R_s^-)^3}\int_0^{R_s^-} r^2c_s^-(r,t)dr. \tag{22}$$

For more details on the SPMeT equations and notation, refer to [20], [25].

B. Minimum time charging problem

The minimum time charging problem is formulated as:

$$\begin{aligned}&\min_{I(t)} \quad t_f \\ &\text{subject to}\\ &\qquad \text{battery dynamics in (17)-(22)}\\ &\qquad V_T(t_0) = V_0,\, T_{\text{cell}}(t_0) = T_0\\ &\qquad SOC_n(t_f) = SOC_{\text{n,ref}},\, I(t) \in \left[I^{\min},\, I^{\max}\right]\\ &\qquad V_T(t) \le V_T^{\max},\, T_{\text{cell}}(t) \le T_{\text{cell}}^{\max}\end{aligned} \tag{23}$$

where $t_0 = 0$ and t_f are the initial and final time of the charging procedure, V_0 and T_0 are the initial value for voltage and temperature respectively, SOC_{ref} is the

reference SOC at which the charging is considered complete. Moreover, $[I_{\min}, I_{\max}]$ is the bound interval for the current while $V_T^{\max}$ and $T_{\text{cell}}^{\max}$, are the upper bounds for voltage and temperature. This is a free-time problem, whose objective is to solve the battery charging problem in minimum time given a battery model and operating constraints. Several publications use this formulation, including [1], [3], [20]. This fast charging problem can be expanded in other forms by modifying the cost function in (23) as maximize the charge throughput over specified time horizon. Work related to this formulation includes [5], [6]. On the other hand, authors in [7], [8], [26] consider SOC reference tracking problem where the cost function (23) is defined as squared difference between the current SOC at time step t and the reference SOC. These formulations fall within the class of state reference tracking problems.

IV. Simulation Results

In this section, we conduct a case study on how the RL framework can be applied to the battery charging problem in simulation. Our goal is to obtain a charging control policy that charges the battery from 0.3 SOC to 0.8 SOC, while the states and outputs do not violate the constraints. We examine the performance of the actor-critic framework for the *minimum time charging* problem using the electrochemical model in Section III. When an electrochemical model is considered, ADP methods become a sensible choice due to the large number of states. We first assume that all the states are available to the agent. Then, we drop this assumption and consider the more realistic scenario in which only temperature and SOC can be measured/computed. Furthermore, we are interested in seeing how actor-critic adapts its learning behavior when the environment changes. This is especially important in battery applications, where the optimal charging trajectory will vary as the battery ages.

In this case study, we consider the minimum charging problem for an electrochemical model, whose chemistry is based on graphite anode/LiNiMnCoO2 (NMC) cathode cell. The PDEs in (17)-(18) are spatially discretized by finite difference. Then, state-space representation is formed with these discretized states and thermal state (20) , which results in a relatively large-scale dynamical systems, 61 states. The actor-critic networks are based on neural network architectures [19] with different numbers of neurons. Specifically, the actor network uses two hidden layers with 20 - 20 neurons. The critic network uses two hidden layers with 100 - 75 neurons. Hyper parameters are detailed in Table I.

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}

Table I: Actor-critic hyper parameters.

The reward function is designed with the aim of both achieving fast charging and guaranteeing safety, according to the optimization problem in (23)

$$r_{t+1} = r_{\text{fast}} + r_{\text{safety}}(\boldsymbol{s}_t, \boldsymbol{a}_t) \tag{24}$$

where $r_{\text{fast}} = -0.1$ is a negative penalty for each time step which passes before the reference SOC is achieved. In addition, a negative penalty is also introduce at each time step in which the voltage and temperature constraints are violated

$$r_{\text{safety}}(\boldsymbol{s}_t, \boldsymbol{a}_t) = r_{\text{volt}}(\boldsymbol{s}_t, \boldsymbol{a}_t) + r_{\text{temp}}(\boldsymbol{s}_t, \boldsymbol{a}_t) \tag{25}$$

This is done in particular by means of linear penalty functions [27]:

$$r_{\text{volt}}(\boldsymbol{s}_t, \boldsymbol{a}_t) = \begin{cases} -100(V_{\text{T}}(t) - V_{\text{T}}^{\max}), & \text{if } V_{\text{T}}(t) \geq V_{\text{T}}^{\max} \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

$$r_{\text{temp}}(\boldsymbol{s}_t, \boldsymbol{a}_t) = \begin{cases} -5(T_{\text{cell}}(t) - T_{\text{cell}}^{\max}), & \text{if } T_{\text{cell}}(t) \geq T_{\text{cell}}^{\max} \\ 0, & \text{otherwise} \end{cases} \tag{27}$$

where constraints are set to $V_{\text{T}}^{\max} = 4.2V$, $T_{\text{cell}}^{\max} = 47°C$ in this case study. The current is limited within the range $[0, 1.8C]$, where C is the C-rate related to the considered cell. The current is applied by scaling and translating the output of the actor network which, in the considered case, is already limited in the range $[-1, 1]$, due to the fact that its last layer is an hyperbolic tangent operator, i.e., $-1 \leq \tanh(\cdot) \leq 1$.

A. Learning Constrained Charging Controls

The objective of this study is to: (i) validate the actor-critic performance on the minimum time charging problem, and (ii) compare the performance with full and reduced state feedback for the actor-critic networks. The performance is measured by the cumulative reward for each episode. In training, the action is determined by following (9) with the presence of exploration noise. In testing, we test the policy without exploration noise so that we can see the performance of the trained actor-critic network.

Figures 3a-b show the training/testing results of the actor-critic approach. The performance of controller during training converges to around -10 cumulative reward while the performance of controller during testing converges to around -5 cumulative reward. The difference comes from the presence of exploration noise. We can clearly observe that exploration is not needed after 1000 episodes as the action network, π, falls into its local optimal. Furthermore, we design two state-feedback controllers[1]. One utilizes the full state vector (61 states) for feedback control. The other uses a reduced or "simplified" state vector for feedback control, with only SOC and temperature (2 states). The purpose of reducing the state vector size is motivated from an intuition that the objective function only involves anode bulk SOC and state constraints. The training/testing results in Fig. 3a-b show that both simplified-states and full-states achieve the goal.

Figures 3c-d show how much constraints are violated during testing. The constraint violation scores are calculated according to $\max\{V_{\text{T}}(t) - V_{\max} \, \forall t \in [0, t_f]\}$, $\max\{T_{\text{cell}}(t) - T_{\text{cell}}^{\max} \, \forall t \in [0, t_f]\}$ for each episode. The constraint violation scores approach zero as the episodes increase, which implies that the controller learns the constraints. Positive values imply the constraints are violated.

[1]Note the state vectors are inputs for both the actor and critic networks

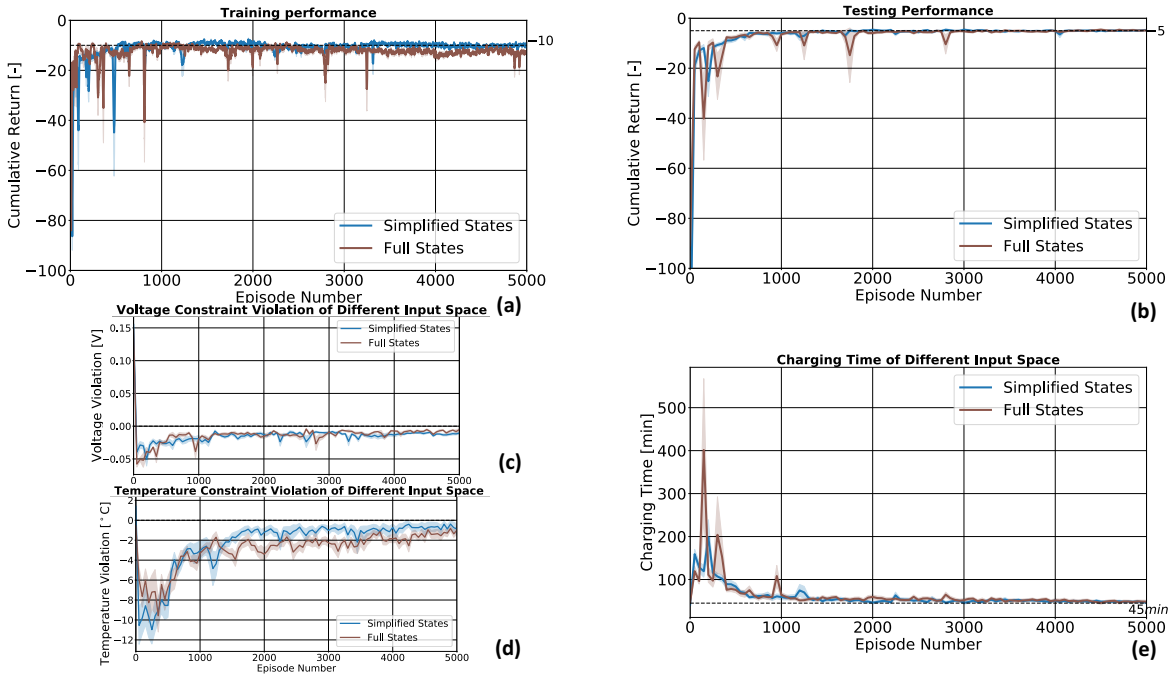

Figure 3: Actor-critic *constrained* charging results for the SPMeT model with 95 % confidence interval. (a) training performance, (b) testing performance, (c,d) constraint violation, (e) charging time

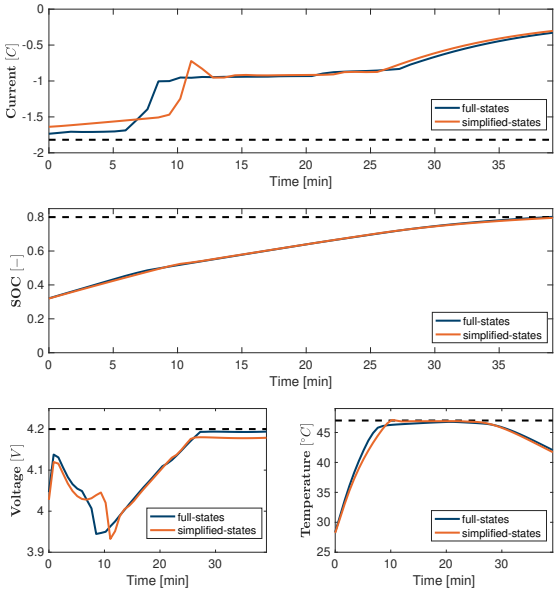

Figure 4: Validation of actor-critic algorithm after training with $V_{\mathrm{T}}(0) = 3.6V, T_{\mathrm{cell}}(0) = 27°C$: (a) simplified-states achieves -5.38 cumulative rewards; (b) full-states achieves -4.69 cumulative rewards.

The constraints are violated in the beginning because parameters of actor-critic are randomly initialized. However, they approach the boundary during learning, since the optimal solution is along the constraint boundary. Figure 3e shows that the charging time decreases to about 45 minutes. Figure 4 visualizes the action, states, and constraints for the simplified-states and full-states actor-critic networks. We find that both controllers achieve similar performance for minimum time charging, around 40 minutes for the given initial conditions. The derived feedback control policy exhibits the constant current (CC), constant temperature (CT), and constant voltage (CV) shape, which can be qualitatively similar to the model-based control results in [1]–[3], [7], [26]. The only difference is that we don't require any knowledge of model dynamics.

B. Learning Adaptive Constrained Charging Controls

In this section, we are interested in the adaptability of the actor-critic approach, which is crucial to the battery charging problem as the cell ages. To represent aging, we perturb the electrochemical parameters, namely, film resistance, $R_f^{\pm}$, heat generation, $Q(t)$. Perturbation of those parameters represents the battery degradation as they directly affect to battery voltage (19) and thermal state (20), which can be monitored by experimental measurement. We expect that the previous actor-critic network could violate the state constraints immediately.

Figures 5a-b display the adaptation results of the actor-critic approach in training/testing. We start from the previous actor-critic configuration in order to observe its adaptability. We observe that both full-sates and simplified actor-critic network are capable of adapting its policy to achieve the goal. We take zoom-in the first 100 episodes to see how the adaptation is processed for the episodes. We observe that the full-states actor-critic network adapts much faster than the simplified states. This is related to the large number of parameters in the full-states network which can lead to greater flexibility in adapting to the new environment.

Figures 5c-d describe the constraint violation scores. Due to change of environment, we observe that the controllers are prone to violate the constraints. Figure 5e shows that the battery charging time increases compared to previous case study because of state violations. The controller reduces the

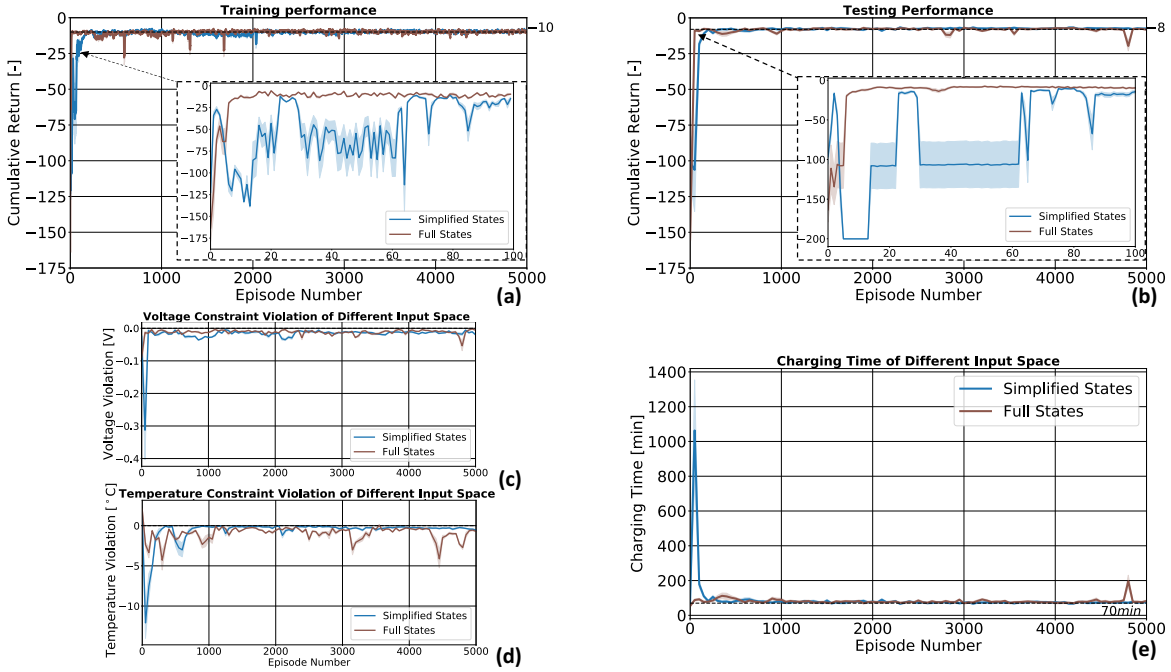

Figure 5: Actor-critic *adaptive* charging results for the SPMeT model with 95 % confidence interval. (a) training performance, (b) testing performance, (c,d) constraint violation, (e) charging time

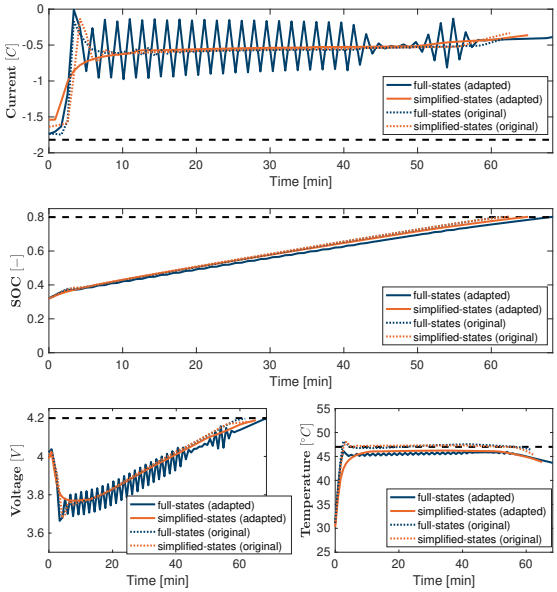

Figure 6: The validation of *adaptive* actor-critic algorithm after training at $V_{\mathrm{T}}(0) = 3.6V, T_{\mathrm{cell}}(0) = 27°C$: (a) *adapted* simplified-states achieves -7.79 cumulative rewards while original simplified-states achieves -81.78 due to state violation; (b) *adapted* full-states achieves -8.19 cumulative rewards while original full-states achieves -82.16 due to state violation.

charging level thus increasing the required charging time in order to reach the reference SOC (0.8) from 45 to 70 minutes.

Figure 6 shows the performance of adaptive controllers at the end of training. The constraints are equivalent to the previous validation, but the system dynamics has changed. So, the original actor-critic network, which learns from a fresh battery, immediately violates the constraints since the environment has changed (aged). However, we are able to construct adaptive controllers for both full-states and simplified-states that achieve the goal without safety violations from previous actor-critic networks. The fluctuating current for the full-states actor-critic network could be mitigated by regularizing the actor-critic parameters during learning.

V. Conclusion

In this paper, we have examined a reinforcement learning approach for the battery fast-charging problem in the presence of safety constraints. In particular, we have shown how RL can overcome many of the limitations of the model-based methods. Among the RL paradigms, the actor-critic paradigm, and specifically the DDPG algorithm, has 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 violation. The control strategy has been tested in simulation on an electrochemical battery model and the presented results are consistent with model-based approaches. In addition, the performance of the actor-critic strategy has been evaluated both in the case of full and partial state feedback. Finally, the adaptability of the control algorithm to battery ageing has been considered. Future work involves adding different types of safety constraints, related to electrochemical phenomena occurring inside the battery, and experimental validation.

VI. Acknowledgements

This research is funded by LG Chem Battery Innovative Contest. The authors thank the LG Chem researchers for their support and discussion in the work.

References

[1] R. Klein, N. A. Chaturvedi, J. Christensen, J. Ahmed, R. Findeisen, and A. Kojic, "Optimal charging strategies in lithium-ion battery", in *American Control Conference (ACC), 2011*, IEEE, 2011, pp. 382–387.

[2] M. Torchio, N. A. Wolff, D. M. Raimondo, L. Magni, U. Krewer, R. B. Gopaluni, J. A. Paulson, and R. D. Braatz, "Real-time model predictive control for the optimal charging of a lithium-ion battery", in *2015 American Control Conference (ACC)*, IEEE, 2015, pp. 4536–4541.

[3] H. E. Perez, X. Hu, S. Dey, and S. J. Moura, "Optimal charging of li-ion batteries with coupled electro-thermal-aging dynamics", *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 7761–7770, 2017.

[4] 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 *2018 European Control Conference (ECC)*, IEEE, 2018, pp. 1753–1758.

[5] B. Suthar, V. Ramadesigan, S. De, R. D. Braatz, and V. R. Subramanian, "Optimal charging profiles for mechanically constrained lithium-ion batteries", *Physical Chemistry Chemical Physics*, vol. 16, no. 1, pp. 277–287, 2014.

[6] B. Suthar, P. W. Northrop, R. D. Braatz, and V. R. Subramanian, "Optimal charging profiles with minimal intercalation-induced stresses for lithium-ion batteries using reformulated pseudo 2-dimensional models", *Journal of The Electrochemical Society*, vol. 161, no. 11, F3144–F3155, 2014.

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

[8] J. Liu, G. Li, and H. K. Fathy, "An extended differential flatness approach for the health-conscious nonlinear model predictive control of lithium-ion batteries", *IEEE Transactions on Control Systems Technology*, vol. 25, no. 5, pp. 1882–1889, 2017.

[9] S. J. Moura, "Estimation and control of battery electrochemistry models: A tutorial", in *2015 54th IEEE Conference on Decision and Control (CDC)*, IEEE, 2015, pp. 3906–3912.

[10] 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", *Industrial & Engineering Chemistry Research*, vol. 58, no. 3, pp. 1286–1299, 2018.

[11] S. Park, D. Kato, Z. Gima, R. Klein, and S. Moura, "Optimal experimental design for parameterization of an electrochemical lithium-ion battery model", *Journal of The Electrochemical Society*, vol. 165, no. 7, A1309–A1323, 2018.

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

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning", *Nature*, vol. 518, no. 7540, p. 529, 2015.

[14] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization", in *International Conference on Machine Learning*, 2015, pp. 1889–1897.

[15] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning", in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[16] F. Belletti, D. Haziza, G. Gomes, and A. M. Bayen, "Expert level control of ramp metering based on multi-task deep reinforcement learning", *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1198–1207, 2017.

[17] 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 Systems*, vol. 32, no. 6, pp. 76–105, 2012.

[18] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning", *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.

[19] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning", *arXiv preprint arXiv:1509.02971*, 2015.

[20] H. Perez, S. Dey, X. Hu, and S. Moura, "Optimal charging of li-ion batteries via a single particle model with electrolyte and thermal dynamics", *Journal of The Electrochemical Society*, vol. 164, no. 7, A1679–A1687, 2017.

[21] D. P. Bertsekas, *Dynamic programming and optimal control*, 3. Athena scientific Belmont, MA, 2005, vol. 1.

[22] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons, 2007, vol. 703.

[23] J. Rust, "Using randomization to break the curse of dimensionality", *Econometrica: Journal of the Econometric Society*, pp. 487–516, 1997.

[24] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms", 2014.

[25] S. J. Moura, F. B. Argomedo, R. Klein, A. Mirtabatabaei, and M. Krstic, "Battery State Estimation for a Single Particle Model with Electrolyte Dynamics", *IEEE Transactions on Control Systems Technology*, vol. 25, no. 2, pp. 453–468, Mar. 2017. DOI: 10.1109/TCST.2016.2571663.

[26] A. Kandel, S. Park, H. Perez, G. Kim, Y. Choi, H. J. Ahn, W. T. Joe, and S. Moura, "Distributionally robust surrogate optimal control for large-scale dynamical systems", in *American Control Conference (ACC), 2020*, IEEE, 2020, To appear.

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