Research Papers

A Reinforcement Learning controller optimizing costs and battery State of Health in smart grids

Marco Mussi [a],*, Luigi Pellegrino [b], Oscar Francesco Pindaro [a], Marcello Restelli [a], Francesco Trovò [a]

[a] *Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milan, 20133, Italy*
[b] *Ricerca Sistema Energetico, Via Rubattino, 54, Milan, 20134, Italy*

ARTICLE INFO

ABSTRACT

Smart Grids are the evolution of traditional electric grids and allow two-way flows of electricity and information between different actors. At the edge of this network, customers can both produce and consume energy. Due to the intermittent nature of renewable energy sources, customers are characterized by moments of energy surplus and deficit. To solve this problem, customers are connected to the power grid, and, usually, they are also provided with Lithium-Ion battery packs positioned near the energy source used to store energy in excess for later use, reducing expensive energy exchanges with the grid. On the one hand, using the battery at its full capabilities produces significant economic savings. On the other hand, massive use of the battery leads to degradation and consequently to a more frequent substitution of the battery. Therefore, depending on the cost of energy and batteries, one should carefully choose when it is favorable to use it. To avoid inefficiencies, it is common to design controllers that regulate the energy flow within the battery packs, deciding whether to exchange energy with the network or store it in the battery. In this work, a Reinforcement Learning controller optimizing energy flow is developed. The controller's goal is to balance the costs of exchanges with the power grid and those derived from the degradation due to battery usage. A synthetic experimental campaign conducted using real-world data demonstrates that the policy learned shows an improvement in the *worst-case* of 3% w.r.t. state-of-the-art baselines.

1. Introduction

In smart grids, Photovoltaic (PV) production becomes a valid and cheap alternative to traditional sources such as fossil fuels [1]. The modularity of this technology allows for the production of energy at different scales, from domestic to industrial use cases. Even if solar energy production has the advantage of being inexhaustible and almost free, it comes with limitations. Indeed, its availability varies highly during the hours of the day, seasons, or due to weather conditions. Furthermore, energy production and peak user demand are often not aligned. To avoid disruptions in the service, all consumers, even if they are also energy producers, are connected to each other and with the traditional power plants through the smart grid to compensate for peaks and eventual lack of energy. The exchanges with the power grid are economically not convenient for the consumers, as the user sells the energy they produce in excess at a usually lower price w.r.t. to the one paid whenever it requires energy from the same network. To overcome this issue, battery packs are usually adopted to store

energy surpluses and meet future demand [2,3]. The use of storage systems enhances the possibility of lowering energy costs and increasing energy independence. On the one hand, a user wants to exploit as much as possible the battery packs to reduce the exchanges with the network which would result in a significant economic expense. On the other hand, extensive use (or misuse) of the battery packs may lead to significant battery degradation, and, in turn, to a more frequent recurring cost due to the substitution of the battery pack. Indeed, battery packs are mainly composed of Lithium-Ions cells, a very efficient and high-energy-density technology, which is affected by a degradation process that lowers their capacity and efficiency over time, caused by the natural aging that each battery incurs, environmental factors (such as storing conditions), and the load applied by the user. The trade-off between use and degradation of the battery is usually addressed by classical controllers, e.g., using simple rules keeping the battery's remaining capacity in a safe range (20%–80%). However, such systems are not optimized to explicitly take into account the data

regarding the cost/price of the energy and the production provided by PV systems, leading to possibly suboptimal strategies to manage energy consumption/production.

Original contribution This work proposes a high-level Reinforcement Learning controller that decides when and how to use the battery to balance between the use of the battery and its degradation. The goal of the controller is to maximize the economic profit from energy production. This task presents several challenges as there are conflicting objectives: a controller should be able to store energy for future uses while avoiding too intensive battery cycling. This work handles this multi-objective reinforcement learning problem by designing a new objective function able to express these conflicting objectives in a unique formulation. This new formulation exploits the data coming from energy production and energy costs experiences in the past to solve the problem using state-of-the-art RL solutions. The hand-crafted MDP model makes use of a well-known degradation model, and the state definition takes into account all the variables that affect the degradation of the battery, as well as other external information to allow the controller to make better usage choices over the lifespan of the battery.[1]

Paper structure The paper starts in Section 2 by discussing the existing literature on the topic and presenting in Section 3 the background on Reinforcement Learning and Lithium-Ion batteries. Then, in Section 4, the problem is formally described, and, in Section 5, a Reinforcement Learning solution to learn the controller's optimal behavior is presented. Subsequently, in Section 6, an experimental validation of the proposed solution w.r.t. to state-of-the-art controllers using real-world users' consumption and production profiles is presented. Lastly, in Section 7, a final discussion of the work is presented, drawing possible research lines to further improve smart grid controllers.

2. Related works

In recent years, a lot of effort has been made to create and integrate Artificial Intelligence (AI)-based methods to solve complex tasks in smart grids. The role of AI and Reinforcement Learning (RL, [5]) in particular is becoming fundamental as smart grids scale their dimension, also due to the challenges in adapting rule-based policies in complex problems [6]. In this section, we present the most relevant works in the field of battery management in smart grids, with particular attention to the ones that make use of reinforcement learning methods.[2]

A line of research is focused on managing the batteries to avoid early degradation. A work closely related to the one presented in this paper is the one proposed by Sui and Song [10], who study the problem of scheduling charge, discharge, and resting periods while using multiple batteries. The proposed scheduler has to keep the State of Charge of every battery above a given level, and at the same time, it has to minimize the degradation caused by high temperatures. It models two different characteristics of a Lithium-Ion battery: *rate capacity effect* and *recovery effect*. Due to the former, a battery shows a smaller overall capacity when discharged at high currents, while the latter influences the battery voltage recovery after a continuous discharge process. The so-designed scheduler properly combines these two effects to extend the battery life. This work considers fixed charge/discharge currents, which simplifies the control problem, but it does not allow the scheduler to choose between different charging or discharging profiles that could achieve the same performance with lower effects in terms of degradation. A shortcoming of this work is that State of Health modeling is influenced only by temperature, and other essential factors such as Depth of Discharge, State of Charge, and current rate are not

considered. Moreover, no economic considerations are done w.r.t. State of Health, and the scheduler's objective is to use a battery for as long as possible while avoiding cycles that generate short-term high degradation.

Another line of research is more focused on the advantages of trading the energy on smart grids and the profit one may get by delivering energy to other users. Huang et al. [11] use an Energy Storage System (ESS) that manages energy produced from renewable sources and introduces an economic criterion to store or deliver energy. Solar and wind energy are characterized by periodic patterns that can be predicted by taking into account meteorological data. This work ties the decision process by predicting the implants' energy produced. The system should follow the energy production profile to store as much energy as possible and sell it when the market conditions are profitable. The controller is designed with an economic perspective: the objective is to maximize the profit by selling energy while keeping into account the operational constraints of the ESS. However, this work does not make considerations about the State of Health of the battery packs that compose the ESS and, therefore, it does not consider the effect of battery degradation on the overall profit. Indeed, one of the simplifying assumptions used is that the ESS has a fixed maximum capacity over time, and the economic effects due to the substitution of the accumulation systems are not considered. Cao et al. [12] design a deep Reinforcement Learning controller that performs energy arbitrage. The objective is to generate profit by storing or releasing energy from an accumulation system, which is bought only to perform arbitration. Indeed, none of the system components produces energy, and therefore the controller takes only into account those exchanges with the electric grid that do not require to produce energy. This approach exploits the past electric market price history to make a prediction for the next 24 h. Then, based on such an estimate, the controller decides which interaction with the grid is the most profitable. The peculiarity of this work is that it considers the effects that battery degradation has only in the profit estimates. The main limitation of this approach is that profit is computed in the short-term (i.e., one week) and, therefore, does not consider profit for long time horizons.

Other works are more focused on the design of energy storage systems that are suitable for the specific setting. For instance, the work by Kell et al. [13] uses a deep Reinforcement Learning controller to regulate energy usage in homes with photovoltaic panels and an accumulation system installed. This technique allows fine-grained control of the current to which a battery is subject, allowing efficient and precise driving. The work aims to find the correct battery size for a given household. The technique is based on the limiting assumption that no relevant battery degradation happens in one year, and the controller needs to be re-trained every time a new battery is considered. Ebell et al. [14] propose a first Multi-Agent Reinforcement Learning approach under partial observability for promoting energy sharing among households. Ebell et al. [15] implement a RL controller with the goal of reducing the exchanges with the power grid for a household equipped with photovoltaic panels and storage systems. Kwon and Zhu [16] uses an RL approach for modeling battery degradation and optimizing economic objectives. In their work, they propose a method to track degradation. However, they do not design a way to keep track of the periodic effects of photovoltaic generation.

3. Background

This section summarizes those technical notions needed to understand the problem. First, in Section 3.1, the Reinforcement Learning background required to understand the problem is presented. Then, in Section 3.2, the background related to the lithium-ion batteries and a commonly adopted model to estimate degradation for such batteries are discussed.

---

[1] A preliminary version of this work first appeared in [4].

[2] For a detailed discussion on this topic, we refer to Zhang et al. [7], Yu et al. [8] and Subramanya et al. [9].

3.1. Reinforcement learning

Reinforcement Learning (RL, [5]) considers an agent which interacts with an environment. The agent is the actor who chooses at each time t the action a_t to perform, and the environment reacts to the agent's actions by evolving its state s_{t+1} and providing a reward, which represents how good it is to perform action a_t in state s_t. The goal of the agent is to maximize the collected rewards. The abstraction used to map and define a Reinforcement Learning problem is the Markov Decision Process (MDP) which allows the formalization of processes with temporal dependencies. Formally, a Markov Decision Process $\mathcal{M}$ is defined as a tuple $\mathcal{M} := (S, \mathcal{A}, P(s'|s, a), R(s, a), \gamma)$, where S is the set of states, $\mathcal{A}$ is the set of actions the controller is allowed to perform, $P(s'|s, a)$ is the state transition probability matrix, $R(s, a)$ is the reward function, and $\gamma \in [0, 1]$ is the discount factor. Given an instance of an MDP $\mathcal{M}$, the goal of a controller is to define a policy $\pi(a|s)$ returning for each state $s \in S$ the action $a \in \mathcal{A}$ which maximizes the discounted sum of future rewards $\sum_{t=1}^{\mathcal{T}} \gamma^{t-1} r_t$, where r_t is the reward, and $\mathcal{T} \in \mathbb{N}$ is the time horizon. The optimal behavior through RL techniques can be learned in two ways. If there is a dataset available, it will be composed of interactions of the form $\{(s_t, a_t, r_t)\}_{t \in \{1, \dots, \mathcal{T}\}}$, where s_t is the state, a_t the action performed, and r_t the instantaneous reward at time t, one can choose *offline* Reinforcement Learning algorithms. If there is no interactions dataset available, one needs an environment (real or simulated) to interact with in order to actively generate a sequence of interactions that can be used for learning. The algorithms that learn while interacting with the environment are called *online* Reinforcement Learning algorithms.

In this work, the authors will focus on Fitted-Q Iteration (FQI, [17]), a value-based algorithm that derives a control policy from batches of transitions previously sampled from the environment in an offline manner. The transitions are sampled with a given policy, whose exploration capabilities will affect the quality of the estimates of the Q-function. A transition is a tuple $\langle s_t, a_t, r_t, s_{t+1} \rangle$, where s_t is starting state of the transition, a_t is the action drawn from the exploratory policy, r_t is the reward obtained by the agent after performing the action a_t in the state s_t, and s_{t+1} is the next state, reached after performing the action a_t in the state s_t.

3.2. Lithium-Ion batteries

Lithium-ion batteries are subject to a degradation process due to calendar and cycle aging. In stationary tasks, e.g., smart grids, the notion of battery health, or State of Health (SoH) at time t is defined as $SoH_t = \frac{C_{t,max}}{C_N}$, where $C_{t,max}$ is the maximum capacity achievable at time t, and C_N is the nominal capacity of the battery. Batteries are subject to a degradation process that lowers their overall capacity over time, and the real capacity quickly moves away from their nominal value. However, such a quantity cannot be measured directly and must be inferred from other measures. How to effectively estimate this quantity is still an open research problem [18–21]. SoH evolution over time is a highly non-linear process caused by irreversible reactions between the anode and the electrolyte, whose dynamics are determined by a large number of factors. Empirical studies (e.g., [22]) showed that most of the degradation is concentrated at the beginning and end of the battery life, while the degradation rate decreases during its mid-life. Fig. 1 shows a qualitative example of the relationship binding the number of cycles and the SoH of the battery.[3]

Degradation model Since the applications using the SoH estimate require frequent updates and high precision, in this work, the authors
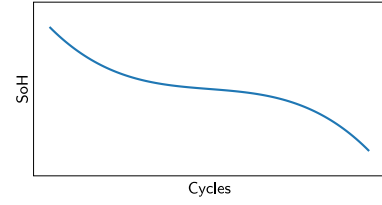

Fig. 1. A qualitative example of the shape of the non-linear relationship that can bind cycle number and SoH [22].

consider a synthetic model to define and keep updated the estimate of the battery degradation. The required precision is difficult to reach using currently available measurement instruments, so the authors choose to keep it updated using a realistic model. In particular, the focus is on the aging model proposed by Xu et al. [23].[4] This model combines theoretical considerations with empirical evidence. Even if the process of the battery degradation is determined by factors such as charging, discharging, time, temperature, and the current state of life, the above-mentioned model assumes that the battery degradation process can be factorized among the time and stress cycle effects. The two factors considered for the degradation model are reflected in the definition of two stress functions: *calendar* and *cycling* aging. Calendar aging is the degradation stress that a battery suffers independently from its use. It depends on the operational lifetime of the battery and other parameters such as the mean State of Charge (SoC, [24–26]), and the mean temperature at which it is preserved. On the other hand, cycling aging is caused by the direct use of the battery. Every cycle is modeled as a single stress event *independent* from the others, and the accumulated degradation is the sum of the capacity reduction caused by each cycle. The overall stress $f_{d,t}$ is a linear combination of calendar and cycling aging. Formally:

$$f_{cal,t} = t \, f_{cal,1}(\bar{\delta}, \bar{\sigma}, \bar{T}),$$

$$f_{cyc,t} = \sum_{i}^{N_C} n_i f_{cyc,1}(\delta_i, \bar{\sigma}_i, \bar{T}_i),$$

$$f_{d,t} = f_{cal,t} + f_{cyc,t},$$

where $\bar{\sigma}$ and $\bar{T}$ are the average State of Charge (SoC) and temperature at which the battery has been stored, respectively, t is the age of the battery, N_C is the number of equivalent cycles, δ_i, $\bar{\sigma}_i$ and T_i are the Depth of Discharge, average State of Charge and Temperature of the ith cycle, respectively, and n_i indicates whether cycle i is a full ($n_i = 1$) of half ($n_i = 0.5$) cycle.

Thanks to the above definitions, the battery degradation is computed as follows:

$$D_t = 1 - \alpha_{sei} e^{-f_{sei,t}} - (1 - \alpha_{sei}) e^{f_{d,t}}, \tag{1}$$

$$f_{sei,t} = \beta_{sei} f_{d,t}. \tag{2}$$

It is worth noting that Eq. (1) suggests that the degradation is non-linear with respect to the overall stress factor $f_{d,t}$. This model reflects that a battery suffers from high degradation rates at the beginning of its life, then it reaches a plateau, and, finally, the degradation increases rapidly when it reaches the end of its life (see Fig. 1). Notice that Eq. (1) represents the battery degradation starting from fresh batteries and does not follow the last phase of battery aging when the capacity rapidly falls. Typically, this latter phase starts when the SoH is proximal to zero. This formulation also considers the fast degradation caused by the SEI, whose formation rate decreases when a stable film has been formed. Therefore, the equation can be seen as divided into two

---

[3] Notice that the degradation D_t can also be used to describe the battery health, i.e., $D_t = 1 - SoH_t$.

[4] A summary of all the quantities considered in the degradation model and the related meaning is provided in Table 1.

Table 1
Table of symbols for the degradation model.

| Symbols	Meaning
t	Generic time
SoH_t	State of Health (SoH) at time t
D_t	Degradation at time t
$C_{t,max}$	Maximum capacity at time t
C_N	Nominal capacity
$f_{d,t}$	Overall aging function at time t
$f_{cal,t}$	Calendar aging function at time t
$f_{cyc,t}$	Cycle aging function at time t
$f_{cal,1}$	Unitary calendar aging function
$f_{cyc,1}$	Unitary cycle aging function
$\bar{\sigma}$	Average (overall) State of Charge
$\bar{T}$	Average (overall) temperature
N_C	Number of equivalent cycles
δ_i	Depth of Discharge (DoD) of cycle i
$\bar{\sigma}_i$	Average State of Charge of cycle i
$\bar{T}_i$	Average temperature of cycle i
n_i	Cycle type indicator of cycle i
$\alpha_{sei}, \beta_{sei}$	SEI coefficients
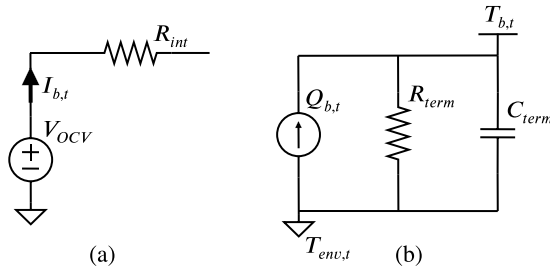
Fig. 2. Electric (a) and thermal (b) models of the battery.

components: one that takes into account the capacity loss caused by the SEI formation, and the other considers capacity fading at a rate proportional to the battery life. Eq. (2) indicates that the SEI formation is proportional to the battery used.

Thermal model The degradation model presented above includes a degradation component related to the temperature at which the current is subject. To include such degradation, a thermal model is needed in order to estimate how the temperature varies over time. More in detail, a thermal model defines how the battery temperature $T_{b,t}$ changes over time. In this scenario, the temperature behavior is controlled by the heat dissipated due to the Joule effect during charges or discharges:

$$Q_{b,t} = I_{b,t}^2 R_{int}, \tag{3}$$

where R_{int} is the internal electric resistance of the battery (see Fig. 2(a)). This effect is a consequence of modeling the battery like a real generator that exhibits a resistive behavior when a current passes through it. The temperature dynamics is modeled with the thermal circuit in Fig. 2(b):

$$L(s) = \frac{R_{term}}{R_{term}C_{term}s + 1}, \tag{4a}$$

$$T_{b,t} = \frac{Q_{b,t}R_{term}\Delta t + T_{b,t-\Delta t}R_{term}C_{term} + T_{env,t}\Delta t}{R_{term}C_{term} + \Delta t}. \tag{4b}$$

Intuitively, Eq. (4a) is the Laplacian transfer function. It describes how heat exchanges happen between the battery and the surrounding environment (in this formulation, $T_{env,t}$ is the environment temperature). Eq. (4b) is the corresponding anti-Laplacian, and it describes how the temperature of the battery changes over time.

4. Problem formulation

This work aims to design a controller optimizing the flow of power in a smart grid. Consider the scheme presented in Fig. 3. The node to
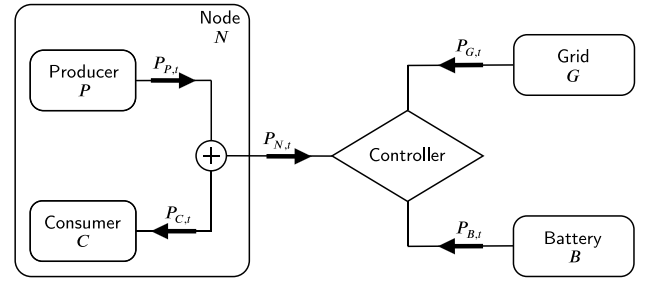

Fig. 3. A schema of the overall structure interacting with the controller. The arrows' directions denote the convention adopted for positive values of the power exchanged by each component.

manage consists of an energy *producer P* (e.g., a photovoltaic panel) and an energy *consumer C* (e.g., a house and its users).

The producer and the consumer asynchronously release and absorb energy, respectively. Consider, for example, the case of a house equipped with photovoltaic panels and with people consuming energy. The panels will produce energy depending on the time of day, season, and weather conditions. Instead, the users will consume energy according to some pattern [27,28]. From now on, the union of producer and consumer will be addressed as a *node N*. During the different periods of the day, the node either needs to have additional energy in input or to manage the energy in excess. To compensate for these consumption peaks and energy lacks, the node is connected to the power *Grid G*, so it can exchange (buy or sell) energy if necessary. However, the process of exchanging energy with the grid follows market logic that is unknown to the node but should be considered. Indeed, energy is bought at a given cost p_{in}, which is significantly higher than the price p_{out} at which energy can be sold to the grid. This condition makes the exchanges disadvantageous for the node. To overcome this problem, *batteries B*, usually Lithium-ion ones, are commonly being adopted as power storage by the nodes to collect and release energy when needed, allowing to reduce the uneconomic exchanges with the grid. In the following, it is assumed that the battery is freshly installed, and it has been bought at the price p_{new}.

Every exchange between the components presented above is measured in terms of exchanged power (denoted as $P_{X,t}$ for a generic component X at time t). A visual representation of the adopted exchange conventions is presented in Fig. 3 (positive values for the powers-related quantities $P_{X,t}$ are denoted by the arrows' directions).[5] Formally, for any time t, the following equations describing the power exchanges hold:

$$\begin{cases} P_{N,t} = P_{P,t} - P_{C,t} \\ P_{N,t} = P_{G,t} + P_{B,t} \end{cases}. \tag{5}$$

This work aims at creating a *controller* capable of optimizing its control policy, minimizing the costs due to exchanges, added to those due to battery degradation. The controller will be designed using the data coming from previous power usage and production profiles and will use Reinforcement Learning techniques to generalize from such data. The task of the controller is, given the amount of power released/absorbed by the node (i.e., $P_{N,t} \in \mathbb{R}$), to decide the percentage of the request that will be satisfied using the battery B, and the those that will be satisfied by the Grid G. The controller monitors these subsystems with a fixed control period Δt for a predetermined number of control steps $\mathcal{T}$. It is assumed that the system has access to the measurement of several physical properties of the battery pack, such as its SoC σ_t, the battery temperature T_t, the DoD of the current cycle δ_t, and the degradation of the battery D_t. The controller can send or request power $P_{G,t}$ from the electric grid generating an economic transaction.

---

[5] A summary of the quantities considered in this section is provided in Table 2.

Table 2
Table of relevant symbols for the formulated problem.

Symbols	Meaning
$P_{X,t}$	Power for a generic component X at time t
$E_{X,t}$	Energy for a generic component X at time t
p_{in}	Price to buy energy from the grid
p_{out}	Price to sell energy to the grid
p_{new}	Price of a new battery

4.1. Controlled variable

The task of the controller is, at every time t, to measure the power coming from the node $P_{N,t}$ (either positive or negative) and decide which fraction $a_t \in [0,1]$ store (retrieve) in (from) the battery. The remaining part of the power is directed to the grid. Formally:

$$P_{B,t} = a_t P_{N,t}, \qquad (6)$$

$$P_{G,t} = (1 - a_t) P_{N,t}, \qquad (7)$$

$$E_{G,t} = P_{G,t} \Delta t, \qquad (8)$$

where $E_{G,t}$ is the energy exchanged with the grid. Note that, according to the notation in Fig. 3, if $P_{B,t}$ is positive, the battery is discharging, and $P_{G,t}$ is positive if the power is requested from the grid. Eq. (8) computes the actual energy exchanged with the grid, and this quantity will be used to compute the economic gain or loss that occurred while interacting with the electric grid.

While operating, the controller must comply with the batteries' constraints, i.e., that each battery charge level cannot be below zero or above the maximum capacity, formally $0 \leq \sigma_t \leq 1$, $\forall t$. In this setting, a second low-level controller is assumed to protect the battery from overcharging or excessive draining by actuating controls on high-level actions.

4.2. Objective

The objective of this work is to find the sequence of actions that reduce as much as possible the amount of money needed to maintain the system and generate profits (reduce losses) while exchanging energy with the grid. The objective comprises two independent elements: the battery cost and energy exchanged with the power grid. The former takes into account the amount of money lost due to battery degradation, and it grows proportionally with degradation. Instead, the latter refers to the profits/losses made when there is an interaction with the electric grid. These two objectives are conflicting since a more aggressive use of the battery could generate favorable trades with the electric network, but it will also increase the degradation rate of the battery. Formally, the goal is to maximize, over the time horizon $\mathcal{T}$:

$$\max_{M} \ R_{total,\mathcal{T}}(M), \qquad (9)$$

where M is a strategy deciding the values of the fractions to store in the battery $(a_0, \dots, a_{\mathcal{T}})$ to perform over the time horizon, and:

$$R_{total,\mathcal{T}}(M) = R_{batt,\mathcal{T}}(M) + R_{exc,\mathcal{T}}(M), \qquad (10)$$

where:

$$R_{batt,\mathcal{T}}(M) = p_{new} \, SoH_{\mathcal{T}}, \qquad (11)$$

$$R_{exc,\mathcal{T}}(M) = \sum_{t=1}^{\mathcal{T}} \Big[p_{out} \left| E_{G,t} \right| \mathbb{1}\{(P_{G,t}) < 0\} - p_{in} \left| E_{G,t} \right| \mathbb{1}\{(P_{G,t}) > 0\} \Big], \qquad (12)$$

where $E_{G,t}$ is the energy exchanged with the Grid G at time t, and $\mathbb{1}\{x\}$ is the indicator functions that is 1 if the condition x is satisfied, and 0 otherwise.

5. Algorithm

In this section, the problem of determining the optimal action of the controller is dealt as a sequential decision problem, where the controller has to find the best sequence of actions that will maximize the objective in Eq. (9). Since the values of the sequence of powers $P_{P,1:\mathcal{T}}$ and $P_{C,1:\mathcal{T}}$ over time are not known in advance, the stochasticity of the environment (e.g., the energy production by PV cells, the daily usage) must be taken into account. The application of optimization techniques would require the full knowledge of such quantities during the entire time horizon of the battery usage. Conversely, the control of the battery actions should be performed before such information is available. Given that, a suitable model for such a setting is the *Markov Decision Process* (MDP), which can describe scenarios in which the environment evolves over time according to stochastic exogenous factors and according to the actions performed by an agent (i.e., in this case, the controller). This model, in combination with power consumption and production data, will be used in the following section to learn a controller using Reinforcement Learning techniques. In the following, the problem of controlling the power consumed/produced is formalized as an MDP, describing the fundamental components of the framework, more specifically, the states, actions, rewards, and discount factor.

5.1. State

First, a specific mapping between the current time and day of the year to an angle on the unit circumference is defined. Indeed, this problem is characterized by two types of periodicity: the day-night periodicity and the seasonal periodicity. This phenomenon can be captured by defining a correspondence between an hour of the day (and day of the year for what concerns the seasonality) and an angle in $[0, 2\pi]$. Formally, the angular position for the time of the day is $\varphi_d = \frac{2\pi \tau_d}{\mathcal{T}_d}$, where $\mathcal{T}_d$ is the number of seconds in a day and $\tau_d \in [0, \mathcal{T}_d]$ is the current second of the day. The angular position for the time of the year is $\varphi_y = \frac{2\pi \tau_y}{\mathcal{T}_y}$, where $\mathcal{T}_y$ is the number of seconds in a year and $\tau_y \in [0, \mathcal{T}_y]$ is the current second of the year.

The state vector at time step t is $s_t \in S \subseteq \mathbb{R}^Q$, defined as follows:

$$s_t = \big(\sigma_t, T_t, \delta_t, P_{N,t}^{rate}, P_{P,t}, \cos(\varphi_{d,t}), \sin(\varphi_{d,t}), \cos(\varphi_{y,t}), \sin(\varphi_{y,t}) \big), \qquad (13)$$

where:

- σ_t is the current battery SoC;
- T_t is the current battery temperature;
- δ_t is the current battery DoD;
- $P_{N,t}^{rate} := \frac{P_{N,t}}{P_B^h}$ is the maximum P-rate that the battery would be subjected to if all the net power $P_{N,t}$ would be directed to the battery, where P_B^h is the power that will discharge the battery in one hour starting from a fully charged battery;
- $P_{P,t}$ is the power generated by the producer;
- $\cos(\varphi_{d,t})$ and $\sin(\varphi_{d,t})$ are the mapping of the angular position for the time of the day to a 2D space;
- $\cos(\varphi_{y,t})$ and $\sin(\varphi_{y,t})$ are the mapping of the angular position for the time of the year to a 2D space.

Some remarks are in order. First, the first four elements of the state s_t directly impact the computation of the degradation and therefore impact the state of the entire node.

Second, $P_{P,t}$ has been included in the MDP state as a proxy of the future sun availability. Indeed, if during the day this value is very low, one can assume that the day is cloudy or rainy, and, therefore, no future power production is also expected in the next hours.[6]

---

[6] The joint use of $P_{P,t}$ and $P_{N,t}^{rate}$ is necessary. Suppose the information about the production $P_{P,t}$ is not provided. In that case, the algorithm will not be able to discern, for example, a situation of no production and no consumption from a situation of high production and high consumption.

Third, the last four components of the state s_t have been used to map the current time into an encoding that is able to express a similarity measure between different periods of the day/year [29].

Finally, the SoH has not been included in the state of the MDP even if the SoH value allows the agent to understand at which point of the degradation curve the battery is. This can create a problem if not correctly managed because, given a fixed behavior, different SoH levels lead to different degradation (see Fig. 1). The value of the SoH is not included because, even if the value of the SoH influences the degradation, it is not relevant to evaluate the gain/loss due to an action on the system. Instead, the SoH has been included in the choice of the model reward, which will be described in Section 5.3.

5.2. Action

An action $a_t \in \mathcal{A} \subseteq [0, 1]$ at time step t consists in the choice (performed by the controller) of the fraction of power $P_{N,t}$ that will be directed to the battery. The remaining power is directed to the grid. The action set $\mathcal{A}$ is either continuous or discrete, and the choice will be influenced by the Reinforcement Learning algorithm that will be adopted. In this work, due to the fact that Fitted-Q Iteration algorithm requires a finite action space (see [17]), a finite action space is selected: $\mathcal{A} := \{a_1, \ldots, a_i, \ldots, a_K\}$, with $a_i \in [0, 1]$.[7]

5.3. Reward

The reward function r_t at a specific time step t, defines the gain/loss of an agent that performed action a_t in state s_t. The reward function is defined as follows:

$$r_t = r_{exc,t} + r_{batt,t}, \tag{14}$$

where:

$$r_{exc,t} = p_{out} |E_{G,t}| \mathbb{1}\{P_{G,t} < 0\} - p_{in} |E_{G,t}| \mathbb{1}\{P_{G,t} > 0\}, \tag{15}$$

and:

$$r_{batt,t} = -\frac{f_{d,t} - f_{d,t-1}}{f_{d,max}} p_{new}. \tag{16}$$

Some remarks are in order. First, the two macro-components of the reward presented in Eq. (14) are the same of the objective function to optimize (Eq. (9)). On the one hand, the component related to the energy exchanges with the grid $r_{exc,t}$ described in Eq. (15) is equal to the one inside the summation in the objective function. On the other hand, in the reward component related to the battery $r_{batt,t}$ (Eq. (16)) there are some minor changes to improve the learning phase of the algorithm. The battery value is amortized by considering the variation in the overall stress $f_{d,t}$, rather than SoH (recall that the SoH is nonlinear over time/cycles, as reported in Fig. 1). By using the linear degradation, the reward is distributed more uniformly over the whole time period (considering a period that ranges among all the battery lifetime), and the agent is still able to understand how much of an impact an action has on the degradation, allowing to agents that maximize the long term profit and able to make a trade-off in profit, also at the beginning of the battery life, when the SoH decrease will be very steep. The degradation is normalized by $f_{d,max}$, the maximum linear degradation value that corresponds to the maximum degradation (i.e., the value of $f_{d,t}$ that when placed in Eq. (1) returns $D_t = 1$).

5.4. Discount factor

The discount factor γ used in this problem has been set to 1, implying that the controller has to be farsighted. The problem can be learned since this formulation considers a finite number of steps $\mathcal{T}$.

5.5. Computational performances

The Computational performances of the algorithm running inside the controller are a problem of paramount importance when one wants to apply RL methods in real-world problems. Indeed, commonly, controllers have to be embedded into computing infrastructures with limited computational performances. Therefore, the goal is to have controllers that are efficient during the prediction phase. It is known that the training phase of the RL controllers is usually costly and requires large computing infrastructures to be performed [5]. However, such a phase is performed only once and can be performed in a different hardware infrastructure than the one available on the storage system. Once the training has been performed the final controller can be moved to the infrastructure used to control the storage system. Commonly, the prediction required to determine the control action is computationally lighter and requires performing operations whose cost is linear in the number of state dimensions Q and in the number of actions K. This allows the use of such techniques even in settings in which the computational power is limited.

6. Experiments

This section presents an experimental campaign in a realistic environment simulated starting from real-world power profiles. The corresponding code, as well as the data used in this section are available at https://github.com/marcomussi/SmartGridController.

6.1. Experimental setting

To test the solution proposed in Section 5, an online simulator of the node has been developed following the OpenAI Gym framework [30]. More specifically, the controller is the agent which interacts with the environment, and the environment models the node's behavior and keeps track of the degradation of the battery, provided by the model presented in Section 3.2. Training and testing have been performed on an *Ubuntu 20.04 LTS server*, equipped with *64x Single Core Intel Xeon (Skylake IBRS)* and *32 GB RAM*.

The environment uses real-world power profiles for both consumers and producers. For what concerns the producer, profiles coming from domestic photovoltaic panels are used, while for the consumer, load profiles of houses are selected. During the simulations, the values of the power profiles are revealed sequentially to the agent, one sample at each time instant. The load profiles include a pool of 398 profiles over 365 days sampled each 3600 s, with a peak consumption of 3 kWh. The producer profile is generated from a pool of 10 over the same period with the same frequency as the load ones, gathered from different power plants.[8] The battery capacity is 8 kWh, and the battery type is LMO. The simulation environment matches a load and consumption profile to have a synthetic scenario simulating both production and consumption.[9] Fig. 4 presents an example of such energy production (blue) and consumption (orange) profiles. In this figure, it is possible to observe different producer profile patterns depending on the weather conditions and different consumption profile patterns due to the week/weekend alternation. The parameters adopted in the simulations regarding the thermal battery model and the degradation model for an LMO battery are reported in Table 3. For what concerns the thermal model (Eq. (4b)), the environment temperature is maintained fixed at 25 °C over time. Instead, the parameters chosen for the degradation models are the ones proposed in [23]. Every episode is run for 8 years, a time in which the battery SoH will reach 0 in the worst-case scenario.

---

[7] Notice that the action prescribed by the controller may differ from the actual one due to the interaction with the low-level controller, which might actuate different actions to avoid dangerous/unfeasible behavior.

[8] Further details on the dataset are provided in [31].

[9] The producer and consumer time series are rescaled to simulate a balanced energy production and consumption system.
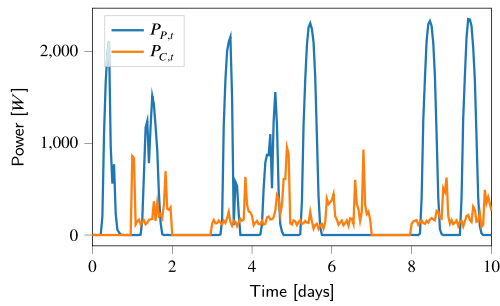
Fig. 4. An example of real-world power profiles (production in blue and consumption in orange) used to generate the synthetic environment. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Parameters of the thermal and degradation model used in the experimental section.

Thermal model		Degradation model	
R_{term}	0.37 °C/W	a_0	38.83
C_{term}	1700 J/K	a_1	7.7
R_{int}	0.005 Ω	a_2	−7.7
T_{env}	25 °C	a_3	9.17
		b_0	−3.51
		b_1	−46

Table 4
Economic quantities considered in the experimental section.

Economic quantities	
p_{new}	375 $/kWh
p_{in}	0.15 $/kWh
p_{out}	0.05 $/kWh

The agent is trained using FQI [17], in the implementation provided by MushroomRL [32]. This algorithm requires the action space to be discrete and a dataset of transitions to learn the optimal policy. Therefore, the agent was allowed to select the action in the set $\mathcal{A} \in \{0.0, 0.1, \ldots, 1.0\}$. The state, reward, and discount factor are set as prescribed in Section 5. To generate the dataset of transitions, a random uniform policy is used. The generated training dataset includes data coming from 100 episodes for a total of 7 million of sampled transitions. FQI is run for 200 iterations using XGBoost [33] as function approximator. The hyperparameters, i.e., the number of iterations, as well as the number of trees, the tree depth, and the minimum number of elements present in a leaf, have been tuned using Optuna [34]. The agent adopts a control period set to $\Delta t = 3600$ s, matching the original data's sampling frequency. Furthermore, the degradation model requires as input the number of cycles in a standardized way, so it requires algorithms like *Rainflow* [35] to quantify cycles in the battery SoC profile. To satisfy this request, an approximated version of Rainflow, called *Streamflow*, has been developed.[10]

The performance of the *RL agent* is compared with 3 baselines:

- *OnlyGrid*: The battery is not used, and all the energy needed (or in excess) is exchanged with the Grid. This corresponds to maintain action: $a_t = 0, \forall t$.[11]
- *OnlyBattery*: The battery is always employed (whenever it is possible). Energy exchanges with the grid happen only when

---

Table 5
Average reward (total, battery, and exchange) after 8 years (10 runs, higher is better).

	$R_{total,\mathcal{T}}$	$R_{batt,\mathcal{T}}$	$R_{exc,\mathcal{T}}$
RL agent	**−2295.32**	−1680.54	−614.78
SoC20/80	−2454.99	−2144.69	−310.30
OnlyBattery	−2365.93	−2141.18	**−224.74**
OnlyGrid	−2354.15	**−1531.66**	−822.49

the battery is completely empty or full. Formally, the action is persisted as: $a_t = 1, \forall t$.[12]
- *SoC20/80*: This baseline keeps the SoC between 0.2 (20%) and 0.8 (80%). This is one of the state-of-the-art control policy [36], since very low or high SoC values are correlated with high degradation. Formally, $\forall t$:

$$a_t = \begin{cases} 1 & \text{if } 0.2 \leq \sigma_t \leq 0.8 \\ 0 & \text{otherwise} \end{cases}. \tag{17}$$

The prices exposed by the electric grid to exchange energy are stationary and fixed at 0.15 $/kWh and 0.05 $/kWh for p_{in} and p_{out}, respectively. The battery acquisition cost $p_{new} = 375$ $ has been set such that the expected reward provided by the *OnlyGrid* and *OnlyBattery* policies are comparable. The costs at which the energy exchanges are performed, as well as the cost to substitute a battery are summarized in Table 4. By leveling off these two policies, the agent has to learn how to optimize the battery's use and energy exchanges, avoiding optimal trivial solutions to the control problem (i.e., always using the battery or not using it at all).[13] Indeed, the choice of selecting the price p_{new} such that the experiments are performed in a balanced situation is made with the purpose of testing the *RL agent* in the most difficult scenario, in order to get a measure of the *worst-case* performance.

The agents are evaluated over several factors to understand the behaviors and the effects of each policy properly. The first evaluation metric for a generic method M at time step t is the total reward $R_{total,t}(M)$. Moreover, two other metrics are taken into account to analyze the behavior of the policies:

- $R_{batt,t}(M)$: the component of the objective function that expresses how much value of the battery was lost while using it (see Eq. (11));
- $R_{exc,t}(M)$: the component of the objective function that sums up the profit made by exchanging energy with the electric grid (see Eq. (12)).

To better analyze the differences between the performances of the agents under analysis, in the following, we report $R'_{x,t}(M)$ the difference in terms of the reward provided by the x component between a method M at time t and the one provided by *SoC20/80*, formally:

$$R'_{x,t}(M) := R_{x,t}(M) - R_{x,t}(SoC20/80),$$

for $x \in \{total, batt, exc\}$. These metrics are collected and averaged over 10 different runs over a time horizon $\mathcal{T}$ corresponding to 8 years.

6.2. Results

Table 5 reports the performances at the end of the time horizon of the *RL agent* and the three baselines. It is worth noting how all the policies are not able to generate a positive profit at the end of

---

[10] The implementation of *Streamflow* is available in the repository.

[11] Using this approach, only the calendar aging impacts the battery degradation. This baseline has the main advantage of preserving the battery. However, the energy exchanges with the grid are always disadvantageous since $p_{out} < p_{in}$.
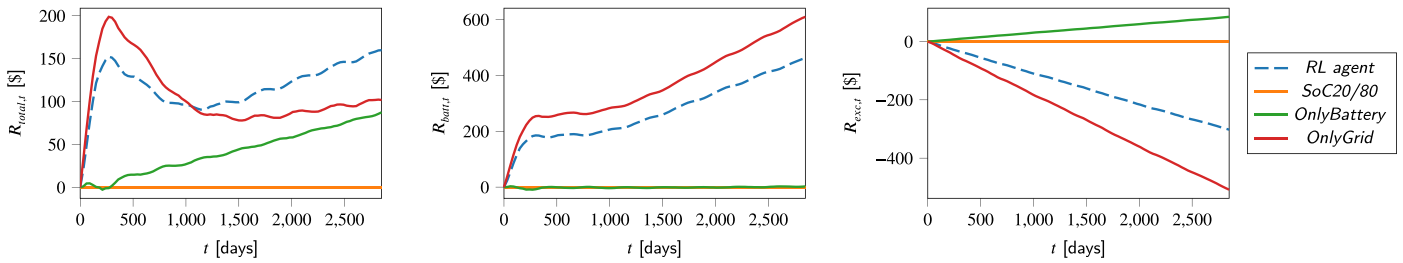
[12] This policy has a significant impact on the battery State of Health, but stores energy for later use, significantly reducing the uneconomic exchanges with the grid.

[13] The value of the expected reward has been estimated by running the two strategies 10 times over a time horizon of 8-years by applying the bisection method.

Fig. 5. Average performance in terms of profit (left), battery cost (center), and energy profit (right) over 8 years experiments, shown by fixing as 0 the results of the *SoC20/80* agent (10 runs, higher is better).

the time horizon. This is due to the energy production, which is not able to satisfy the needs of the node, and that the prices of energy are such that $p_{in} > p_{out}$, which gives no space for economic speculation on energy trading. As expected, the *OnlyGrid* and the *OnlyBattery* are achieving similar results in terms of $R_{total,\mathcal{T}}(M)$. Even if they are the best strategies in terms of use of the battery ($R_{batt,\mathcal{T}}(M)$) and use of the grid ($R_{exc,\mathcal{T}}(M)$), respectively, their total performance is below the one of *RL agent*. Indeed, the *RL agent* provides an average improvement of about 3% over the aforementioned strategies. Finally, in this setting, the *SoC20/80* strategy provides the worse performance, performing worst of the *RL agent* for about 7%.

Fig. 5 presents the performance $R'_{x,t}(M)$ of the analyzed strategies over the 8 years spanning the experiment. *SoC20/80* performs worse than the other policies over almost the entire time horizon in terms of total reward. This is due to the fact that it degrades the battery heavily, similar to what *OnlyBattery* does, without taking advantage of such massive use of the power storage component of the node. The *RL agent* and the *OnlyGrid* agent accumulate most of the advantage w.r.t. *SoC20/80* in the first 200 days of the experiment (corresponding to the peak in Fig. 5, left), while subsequently, the *RL agent* is able to limit the drawbacks of this early improvement. Indeed, looking at Fig. 5, center, we see that for $t > 200$ days the *RL agent* starts relying less on the battery, which leads to an overall larger total reward at the end of the time horizon. Finally, the *OnlyBattery* approach achieves a linear improvement over the *SoC20/80* strategy, but its instantaneous improvements are less effective than the ones provided by *RL agent* and *OnlyGrid*. Indeed, the total reward is smaller for the entire time horizon than the other two. We remark that comparing the behavior of the *RL agent* and *OnlyBattery* total reward $R'_{total,t}(M)$ for $t > 1200$ days they are similar in terms of performance, i.e., the two corresponding lines in Fig. 5, left, are parallel. However, such a performance is achieved using different strategies, since the *RL agent* uses a mix of the battery and grid, while the *OnlyBattery* does not make use of the possibility of exchanging energy with the grid. This further strengthens the idea that building a controller able to manage in an optimized way these two components provides a significant improvement to the management of the smart grid node.

7. Conclusions and future works

Photovoltaic panels are used in residential environments to produce cheap and clean energy, lowering electricity costs and increasing energy independence. Profit is generated by meeting the demand, thus avoiding expensive energy exchanges with the energy grid. The main difficulties in managing such systems are caused by the unpredictable nature of solar energy production and by the asynchronicity between energy production and consumption. To alleviate these limitations, accumulation systems are used to store energy in excess for later use. However, Lithium-ion batteries, are characterized by a process degradation influenced by environmental factors and dynamic loading. This work presents a Reinforcement Learning controller considering a degradation model that allows computing the instantaneous SoH loss. The objective is to maximize the long-term profit while exchanging

energy with the electric grid and by amortizing the battery cost for the whole period according to its use. The proposed algorithm outperforms the state-of-the-art all the baselines of 3% in the *worst-case* scenario where there is the balanced situation described in Section 6. The problem at hand is a combination of multiple complex sub-problems, such as solar energy prediction, electricity prices prediction, energy arbitrage, and degradation modeling. A series of simplifying hypotheses are performed in this seminal work. A first future work can take into account non-stationary energy prices. Efficient energy arbitrage is possible to achieve when there are price fluctuations that can be exploited to make a profit.

CRediT authorship contribution statement

Marco Mussi: Conceptualization, Methodology, Visualization, Writing – original draft. **Luigi Pellegrino:** Conceptualization, Data curation, Project administration, Supervision, Writing – review & editing. **Oscar Francesco Pindaro:** Software, Writing – original draft, Data curation, Visualization. **Marcello Restelli:** Conceptualization, Project administration, Supervision, Writing – review & editing. **Francesco Trovò:** Conceptualization, Project administration, Writing – review & editing, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

GitHub link available in the paper.

Acknowledgments

References

[1] E. Kabir, P. Kumar, S. Kumar, A.A. Adelodun, K.-H. Kim, Solar energy: Potential and future prospects, Renew. Sustain. Energy Rev. 82 (2018) 894–900.

[2] S.-C. Choi, M.-h. Sin, D.-R. Kim, C.-Y. Won, Y.-C. Jung, Versatile power transfer strategies of PV-battery hybrid system for residential use with energy management system, in: International Power Electronics Conference, 2014, pp. 409–414.

[3] S. Skander-Mustapha, I. Slama-Belkhodja, Energy management of rooftop PV system including battery storage: Case study of ENIT building, in: International Conference on Electrical and Information Technologies, 2020, pp. 1–6.

[4] O.F. Pindaro, Controlling lithium-ion batteries through reinforcement learning, Politecnico di Milano, 2022.

[5] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, 2018.

[6] D. Vamvakas, P. Michailidis, C. Korkas, E. Kosmatopoulos, Review and evaluation of reinforcement learning frameworks on smart grid applications, Energies 16 (14) (2023) 5326.

[7] D. Zhang, X. Han, C. Deng, Review on the research and practice of deep learning and reinforcement learning in smart grids, CSEE J. Power Energy Syst. 4 (3) (2018) 362–370.

[8] L. Yu, S. Qin, M. Zhang, C. Shen, T. Jiang, X. Guan, A review of deep reinforcement learning for smart building energy management, IEEE Internet Things J. 8 (15) (2021) 12046–12063.

[9] R. Subramanya, S.A. Sierla, V. Vyatkin, Exploiting battery storages with reinforcement learning: a review for energy professionals, IEEE Access 10 (2022) 54484–54506.

[10] Y. Sui, S. Song, A multi-agent reinforcement learning framework for lithium-ion battery scheduling problems, Energies 13 (8) (2020) 1982.

[11] S. Huang, P. Li, M. Yang, Y. Gao, J. Yun, C. Zhang, A control strategy based on deep reinforcement learning under the combined wind-solar storage system, IEEE Trans. Ind. Appl. (2021).

[12] J. Cao, D. Harrold, Z. Fan, T. Morstyn, D. Healey, K. Li, Deep reinforcement learning-based energy storage arbitrage with accurate lithium-ion battery degradation model, IEEE Trans. Smart Grid 11 (5) (2020) 4513–4521.

[13] A.J. Kell, A.S. McGough, M. Forshaw, Optimizing a domestic battery and solar photovoltaic system with deep reinforcement learning, 2021, arXiv preprint arXiv:2109.05024.

[14] N. Ebell, M. Gütlein, M. Pruckner, Sharing of energy among cooperative households using distributed multi-agent reinforcement learning, in: 2019 IEEE PES Innovative Smart Grid Technologies Europe, ISGT-Europe, IEEE, 2019, pp. 1–5.

[15] N. Ebell, F. Heinrich, J. Schlund, M. Pruckner, Reinforcement learning control algorithm for a pv-battery-system providing frequency containment reserve power, in: 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, SmartGridComm, IEEE, 2018, pp. 1–6.

[16] K.-b. Kwon, H. Zhu, Reinforcement learning-based optimal battery control under cycle-based degradation cost, IEEE Trans. Smart Grid 13 (6) (2022) 4909–4917.

[17] D. Ernst, P. Geurts, L. Wehenkel, Tree-based batch mode reinforcement learning, J. Mach. Learn. Res. 6 (2005) 503–556.

[18] L. Ungurean, G. Cârstoiu, M.V. Micea, V. Groza, Battery state of health estimation: a structured review of models, methods and commercial devices, Int. J. Energy Res. 41 (2) (2017) 151–181.

[19] R. Xiong, L. Li, J. Tian, Towards a smarter battery management system: A critical review on battery state of health monitoring methods, J. Power Sources 405 (2018) 18–29.

[20] Y. Wang, J. Tian, Z. Sun, L. Wang, R. Xu, M. Li, Z. Chen, A comprehensive review of battery modeling and state estimation approaches for advanced battery management systems, Renew. Sustain. Energy Rev. 131 (2020) 110015.

[21] M. Mussi, L. Pellegrino, M. Restelli, F. Trovò, An online state of health estimation method for lithium-ion batteries based on time partitioning and data-driven model identification, J. Energy Storage 55 (2022) 105467.

[22] R. Spotnitz, Simulation of capacity fade in lithium-ion batteries, J. Power Sources 113 (1) (2003) 72–80.

[23] B. Xu, A. Oudalov, A. Ulbig, G. Andersson, D.S. Kirschen, Modeling of lithium-ion battery degradation for cell life assessment, IEEE Trans. Smart Grid 9 (2) (2016) 1131–1140.

[24] J. Chiasson, B. Vairamohan, Estimating the state of charge of a battery, in: Proceedings of the American Control Conference, Vol. 4, IEEE, 2003, pp. 2863–2868.

[25] R. Xiong, J. Cao, Q. Yu, H. He, F. Sun, Critical review on the battery state of charge estimation methods for electric vehicles, IEEE Access 6 (2017) 1832–1843.

[26] M. Mussi, L. Pellegrino, M. Restelli, F. Trovò, A voltage dynamic-based state of charge estimation method for batteries storage systems, J. Energy Storage 44 (2021) 103309.

[27] T. Logenthiran, D. Srinivasan, T.Z. Shun, Demand side management in smart grid using heuristic optimization, IEEE Trans. Smart Grid 3 (3) (2012) 1244–1252.

[28] W. Kong, Z.Y. Dong, Y. Jia, D.J. Hill, Y. Xu, Y. Zhang, Short-term residential load forecasting based on LSTM recurrent neural network, IEEE Trans. Smart Grid 10 (1) (2017) 841–851.

[29] A. Adams, P. Vamplew, Encoding and decoding cyclic data, South Pac. J. Nat. Sci. 16 (1998) 54–58.

[30] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, W. Zaremba, Openai gym, 2016, arXiv preprint arXiv:1606.01540.

[31] D. Fioriti, L. Pellegrino, G. Lutzemberger, E. Micolano, D. Poli, Optimal sizing of residential battery systems with multi-year dynamics and a novel rainflow-based model of storage degradation: An extensive Italian case study, Electr. Power Syst. Res. 203 (2022) 107675.

[32] C. D'Eramo, D. Tateo, A. Bonarini, M. Restelli, J. Peters, MushroomRL: Simplifying reinforcement learning research, J. Mach. Learn. Res. 22 (131) (2021) 1–5.

[33] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 785–794.

[34] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019.

[35] M. Matsuishi, T. Endo, Fatigue of Metals Subjected to Varying Stress, Vol. 68, Japan Society of Mechanical Engineers, Fukuoka, Japan, 1968, pp. 37–40.

[36] J. Jiang, W. Shi, J. Zheng, P. Zuo, J. Xiao, X. Chen, W. Xu, J.-G. Zhang, Optimized operating range for large-format LiFePO4/graphite batteries, J. Electrochem. Soc. 161 (3) (2013) A336.