energies

MDPI

Article

Extending the BESS Lifetime: A Cooperative Multi-Agent Deep Q Network Framework for a Parallel-Series Connected Battery Pack

Nhat Quang Doan , Syed Maaz Shahid , Tho Minh Duong , Sung-Jin Choi and Sungoh Kwon *

Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan 44610, Republic of Korea; nquang.doan21@gmail.com (N.Q.D.); maaz.shahid26@gmail.com (S.M.S.); duongthominhdtv50@gmail.com (T.M.D.); sjchoi@ulsan.ac.kr (S.-J.C.)

* Correspondence: sungoh@ulsan.ac.kr

Abstract: In this paper, we propose a battery management algorithm to maximize the lifetime of a parallel-series connected battery pack with heterogeneous states of health in a battery energy storage system. The growth of retired lithium-ion batteries from electric vehicles increases the applications for battery energy storage systems, which typically group multiple individual batteries with heterogeneous states of health in parallel and series to achieve the required voltage and capacity. However, previous work has primarily focused on either parallel or series connections of batteries due to the complexity of managing diverse battery states, such as state of charge and state of health. To address the scheduling in parallel-series connections, we propose a cooperative multi-agent deep Q network framework that leverages multi-agent deep reinforcement learning to observe multiple states within the battery energy storage system and optimize the scheduling of cells and modules in a parallel-series connected battery pack. Our approach not only balances the states of health across the cells and modules but also enhances the overall lifetime of the battery pack. Through simulation, we demonstrate that our algorithm extends the battery pack's lifetime by up to 16.27% compared to previous work and exhibits robustness in adapting to various power demand conditions.

Keywords: battery energy storage system; retired lithium-ion batteries; battery management system; deep reinforcement learning; multi-agent; cooperative learning

check for updates

Citation: Doan, N.Q.; Shahid, S.M.; Duong, T.M.; Choi, S.-J.; Kwon, S. Extending the BESS Lifetime: A Cooperative Multi-Agent Deep Q Network Framework for a Parallel-Series Connected Battery Pack. *Energies* **2024**, *17*, 4604. https://doi.org/10.3390/en17184604

Academic Editor: Huang Zhang

Received: 23 July 2024
Revised: 27 August 2024
Accepted: 11 September 2024
Published: 13 September 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Lithium-ion batteries used in electric vehicles (EVs) have a finite lifetime, and for safety reasons must be replaced when their capacity drops to 80% or below [1]. However, these batteries can still be repurposed for other applications, where the remaining capacity is sufficient for less demanding tasks. Figure 1 illustrates the life cycle of lithium-ion batteries, where a battery energy storage system (BESS) can effectively utilize retired batteries when their state of health (SOH) is between 80% and 60% [2,3]. The BESS comprises one or more battery packs, each of which uses a group of battery cells connected in parallel and in series [4], to store electrical energy as backup power for households, data centers, charging stations, etc. [5]. Battery cells in a battery pack can be connected in one of two architectures shown in Figure 2: (a) a module groups the batteries in series, and then the modules are connected in parallel (denoted S-P), which is useful for high voltage applications; and (b) a module groups batteries in parallel, and then the modules are connected in series (denoted P-S) for applications requiring high capacity. In the past few years, many projects around the world have implemented a BESS by repurposing EV batteries. In the Netherlands, a 2.8 MWh BESS was installed for Johan Cruijff Arena in 2018 by reusing Nissan LEAF battery packs, each consisting of 192 cells, in an S-P connection [6]. In Finland, a 2.6 MWh BESS was built in 2021 as a backup power resource for the power grid by repurposing Tesla Model S battery packs, each consisting of 7104 cells, in a P-S connection [7]. Table 1 summarizes recent projects that repurposed EV batteries for a BESS. By leveraging a BESS, the demand for new batteries can be significantly reduced, thereby lessening the environmental impact associated with battery production. However, a BESS that reuses retired lithium-ion batteries from EVs still has some limitations when implemented.

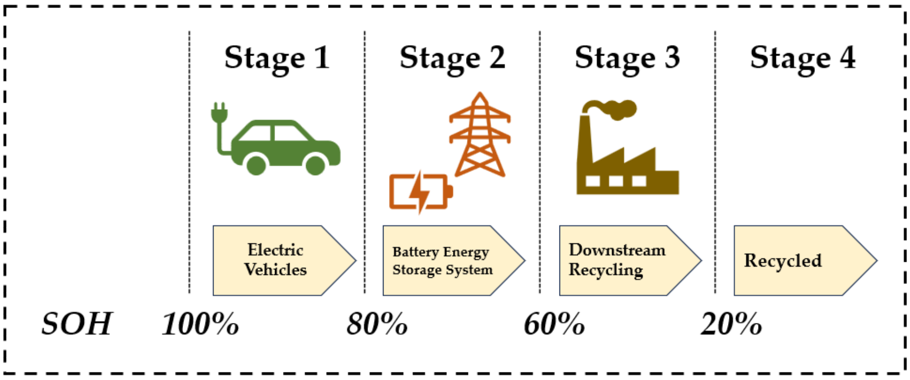


Figure 1. Full life cycle of lithium-ion batteries.

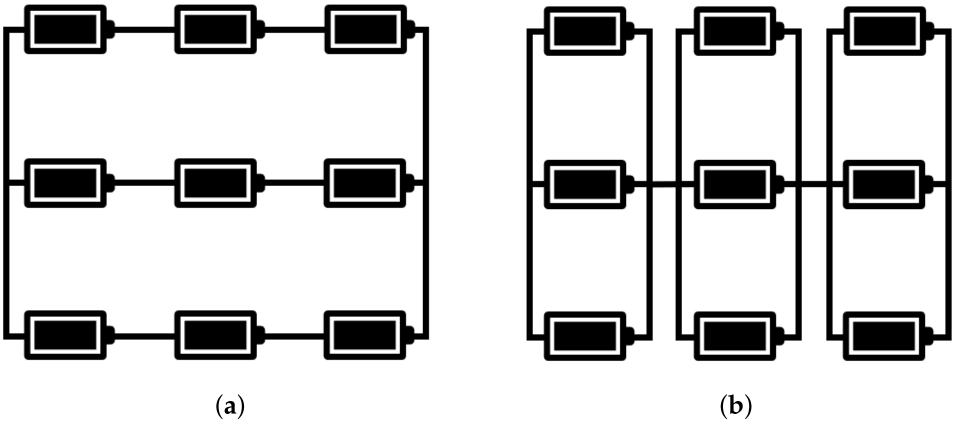


Figure 2. Connections in a battery pack: (**a**) series-parallel (S-P), and (**b**) parallel-series (P-S).

One significant challenge in implementing a BESS is the varying capacity levels of the repurposed batteries connected in parallel and in series, indicating heterogeneous SOHs. The cells within a retired battery pack that exhibits heterogeneous SOHs can result in capacity and voltage imbalances, leading to inefficient energy storage and distribution [8]. This discrepancy in capacity can cause weaker cells to discharge faster or to overheat, potentially shortening the lifetime of the entire battery pack and posing safety risks. Furthermore, the varying SOHs among the cells complicates the task of maintaining balanced charging or discharging across the battery pack, making it difficult to achieve an optimal performance and lifetime [9]. As a result, the cells and modules need to be appropriately connected or bypassed (known as *scheduling*) to mitigate these issues and ensure the reliable operation of a BESS that reuses EV batteries.

Table 1. Projects around the world that reused batteries from EVs.

Name's Project	Applications	Capacity	EV Model	Battery Pack Configuration
Johan Cruijff Arena (Netherlands) [6]	PV power supply, emergency supply	2.8 MWh	590 Nissan LEAF battery packs	96S-2P [1] (192 cells)
Former coal-fired power plant in Elverlingsen (Germany) [7]	Energy storage system for households	3.0 MWh	72 Renault Zoe battery packs	96S-2P (192 cells)
Cactos One Energy Storages (Finland) [10]	Energy storage system for households	2.6 MWh	Tesla Model S battery packs	74P-96S [2] (1704 cells)
EUREF Campus (Germany) [11]	Multi-use storage unit compensates for fluctuations in the grid	1.9 MWh	Audi battery packs	4P-108S (432 cells)
TGN Energy battery energy storage (Norway) [12]	Increased self-consumption	216 kWh	Mercedes-Benz battery packs	NA
Landafors hydropower plant (Sweden) [13]	Offers fast frequency reserve regulation to the power markets	250 kWh	48 Volvo plug-in hybrid battery packs	NA

[1] Ninety-six series-connected batteries in a module, then two modules are connected in parallel. [2] Seventy-four parallel-connected batteries in a module, then 96 modules are connected in series.

In the BESS, switches are integrated to schedule cells and modules by connecting or bypassing them [14]. These switches enable scheduling to selectively isolate degraded cells or modules, thereby extending the battery pack's useful lifetime [15]. By dynamically adjusting the connections between cells and modules, it is possible to balance the load effectively and mitigate the impact of lower capacity cells and modules. Scheduling not only improves the reliability and efficiency of the BESS, but also reduces the need for new batteries by maximizing the utilization of existing battery resources. To achieve this, scheduling requires all the states in the BESS including the state of charge (SOC) and the SOH of the cells and modules, the terminal voltage and output current of the battery pack, the power demand (e.g., from households), and the available power supply (e.g., from solar energy) [16–18]. The SOC of a battery indicates the current charge level as a percentage of its maximum capacity, whereas the SOH represents the ratio of the battery's current maximum capacity to its original rated capacity. The required power demand and available power supply, collectively known as external systems information, refer to the amount of power the BESS needs to discharge and recharge. Incorporating BESS states into scheduling policy protects the battery pack by preventing excessive charging or discharging, and helps connect or bypass cells and modules appropriately. Furthermore, scheduling balances the SOC and SOH across cells and modules, ensuring that no single cell or module is overloaded [19]. In particular, SOH balancing through scheduling reduces the difference between SOHs among cells by utilizing the cells with higher SOHs and bypassing the cells with the lowest SOHs. In this way, the rate of degradation of cells or modules with lower SOHs is minimized. This balance helps to distribute the load more evenly, further protecting the battery pack and extending its useful lifetime.

Scheduling for battery packs in a BESS has been explored in the literature. In [20], an adaptive control algorithm was proposed to balance the SOCs for a series-connected battery pack. However, this approach overlooked the SOH and did not address parallel connections, limiting its effectiveness in comprehensive battery management. In [21], a controller was focused on balancing the SOCs in parallel-connected batteries to prevent overcharge or overdischarge, but failed to consider SOHs and series connections. An approach was presented in [15], where SOC balancing in parallel-series connections was proposed for automatic configuration according to the dynamic load, the storage demand, and the condition of each cell (i.e., SOC and current), yet it ignored SOH, which impacts a BESS lifetime. Other methods proposed in [18,22] were aimed at balancing SOHs by adjusting the charge and discharge durations for cells with weaker SOHs, but they ignored SOC and external systems information. Traditional methods like those presented in [15,18,20–22] are limited when they do not consider both SOC and SOH, because ignoring one can lead to suboptimal performance and reduced battery lifetime [19].

Deep reinforcement learning (DRL) has become a promising direction for battery pack scheduling with its ability to observe multiple states in a BESS and develop appropriate scheduling policies to optimize problem formulation. The combination of neural networks with reinforcement learning in DRL has proven to be a significant breakthrough, enabling the development of more scalable and efficient battery management strategies [23]. Unlike traditional reinforcement learning, which struggles with high-dimensional state spaces, DRL can leverage neural networks to approximate value functions and policies more effectively [24], allowing it to manage a larger number of cells and modules while responding in real time to critical factors such as SOC, SOH, and external systems information. In [17], an SOH balancing framework was proposed based on DRL in a series-connected battery pack to minimize the SOH imbalance among battery cells by observing the cell SOCs and SOHs. However, this approach lacked the observation of factors like power demand, terminal voltage, and output current, which are essential for effective switch scheduling and battery pack longevity. In [16], the authors proposed a DRL-based battery management algorithm to maximize the lifetime of retired batteries with varying SOHs in a parallel-connected battery pack, but they ignored scheduling for series-connected modules, so the approach was limited to applications requiring higher voltages. Moreover, the computational complexity in [16,17] was relatively high due to the extensive state space considered, and the proposals were limited to the use of a single agent, which can lead to a struggle with scalability when there is a large number of cells and modules. In [25], a multi-agent DRL-based method was proposed to reduce the SOC and SOH imbalance among battery cells, but overlooked external systems information, which directly affects battery pack lifetime by preventing overcharging or discharging, especially for cells or modules with lower SOHs. Table 2 shows the classification among battery scheduling algorithms.

Table 2. Classification of battery scheduling algorithms.

References	**Battery Connection**			**SOC Balancing**	**SOH Balancing**	**Lifetime Optimal**	**Dynamic Power Demand**
	Parallel	**Series**	**Parallel-Series**				
[20]	-	✓	-	✓	-	-	-
[21]	✓	-	-	✓	-	-	-
[15]	✓	✓	✓	✓	-	-	✓
[18,22]	-	✓	-	✓	✓	-	-
[17]	✓	✓	-	✓	✓	-	-
[16]	✓	-	-	✓	✓	✓	✓
[25]	✓	-	-	✓	✓	-	-
Proposed	✓	✓	✓	✓	✓	✓	✓

In this study, we propose a battery management algorithm to maximize the BESS lifetime in a parallel-series connected battery pack with heterogeneous SOHs. To carry this out, the proposed algorithm first estimates the SOC and SOH of all cells jointly online. Then, based on the SOCs and SOHs of the cells and modules, a cooperative multi-agent deep Q network framework is implemented to schedule switches in the parallel-series connected battery pack by connecting or bypassing battery cells and modules. The proposed algorithm maximizes the battery pack lifetime and reduces the SOH imbalance among cells and modules. The algorithm also adapts to changes in external systems (i.e., power demand and available power supply). We demonstrate the effectiveness of our proposed algorithm via simulation using real, measured data compared to previous work.

The rest of this paper is organized as follows: Section 2 explores the proposed parallel-series connected battery pack and the associated scheduling challenges. Section 3 formulates the optimization problem by minimizing the reduction in SOH in the battery pack. Section 4 presents the framework of the proposed algorithm. Section 5 details the simulation setup, results, and the algorithm's impact. Finally, Section 6 provides the conclusion of this work.

2. System Model

In this paper, we consider a parallel-series connected BESS [15] with a power supply (e.g., a solar energy generator) and a load (e.g., a household), as shown in Figure 3. The BESS comprises a parallel-series connected battery pack and a battery management system (BMS) that controls charging and discharging. We consider a discrete-time model with time slot t ($t = 0, 1, 2, \ldots$) and durations of Δt.

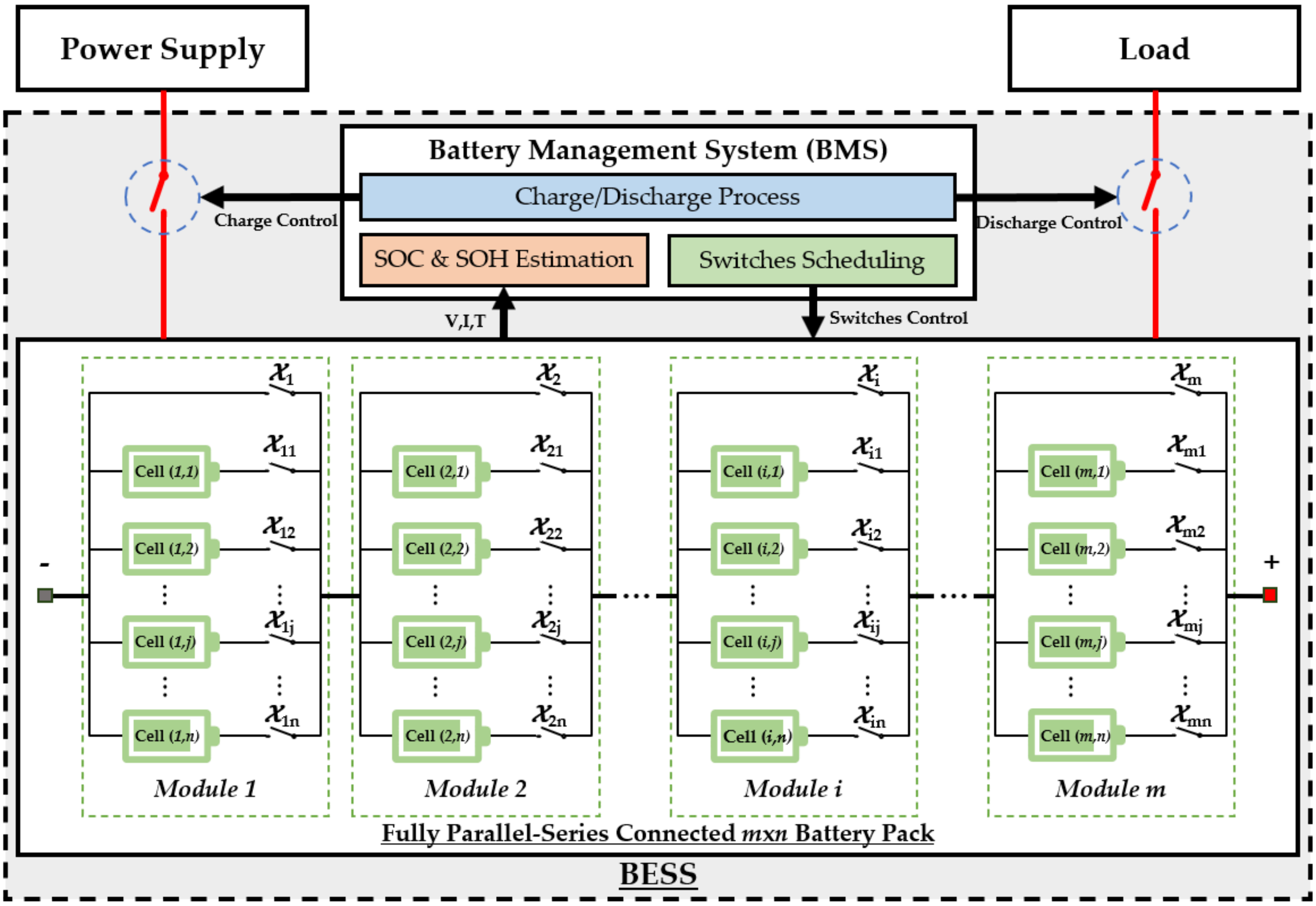


Figure 3. An example of a parallel-series connected BESS.

2.1. Parallel-Series Connected Battery Pack Model

The battery pack consists of $m \times n$ battery cells, where m battery modules are connected in series to increase voltage at the battery pack terminals, and n battery cells are connected in parallel to form a battery module to provide a higher current (or capacity), as shown in Figure 3. To schedule modules and cells in the battery pack, we consider $m \times (n+1)$ controllable switches. Specifically, at time slot t, switch $\mathcal{X}_{ij}(t)$ $\big(i = \{1, 2, \ldots, m\}$, $j = \{1, 2, \ldots, n\}\big)$ expresses connecting or disconnecting battery cell j of module i, denoted as cell (i, j), and switch $\mathcal{X}_i(t)$ is used to express connecting or bypassing module i in the battery pack. Switch $\mathcal{X}_{ij}(t)$ for cell (i, j) at time slot t is defined as

$$\mathcal{X}_{ij}(t) = \begin{cases} 1, & \text{if cell } (i,j) \text{ is connected} \\ 0, & \text{if cell } (i,j) \text{ is bypassed.} \end{cases} \tag{1}$$

Switch $\mathcal{X}_i(t)$ for module i at time slot t is defined as

$$\mathcal{X}_{i}(t) = \begin{cases} 1, & \text{if } \mathcal{X}_{ij}(t) = 0 \text{ for } \forall j \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Switch $\mathcal{X}_i(t)$ is turned ON (1) if all n cells in module i are disconnected (0) to ensure that the battery pack charging or discharging process is not interrupted, which means module i is bypassed from the power supply or load at time slot t. Otherwise, switch $\mathcal{X}_i(t)$ is turned OFF (0) if any cell in module i is connected (1), which means that module i can be charged or discharged at time slot t.

Each battery cell is modeled according to a second-order Thévenin equivalent model [26] based on the material structure of the lithium-ion battery [25], as shown in Figure 4. Battery cell (i, j), which represents the jth cell in module i, has electrochemical impedance spectroscopy (EIS) parameters including open-circuit voltage V_{ij}^{O}, internal resistance R_{ij}^{S}, and two polarization RC pairs connected in series; each RC pair includes a resistor and a capacitor connected in parallel, i.e., (R_{ij}^{P1},C_{ij}^{P1}) and (R_{ij}^{P2},C_{ij}^{P2}). The terminal voltage of cell (i, j) at time t, $V_{ij}(t)$, is calculated by

$$V_{ij}(t) = V_{ij}^{O}(t) - V_{ij}^{P1}(t) - V_{ij}^{P2}(t) - R_{ij}^{S}(t)I_{ij}(t), \tag{3}$$

where $I_{ij}(t)$ is the measured current of cell (i, j) at time slot t; $V_{ij}^{P1}(t)$ and $V_{ij}^{P2}(t)$, respectively, are the polarization voltages of RC pairs (R_{ij}^{P1},C_{ij}^{P1}) and (R_{ij}^{P2},C_{ij}^{P2}) in cell (i, j) at time slot t, and are calculated by using the EIS parameters at the previous time slot $t-1$ as [26]

$$V_{ij}^{P1}(t) = e^{-\frac{\Delta t}{R_{ij}^{P1}(t-1)C_{ij}^{P1}(t-1)}} V_{ij}^{P1}(t-1) + R_{ij}^{P1}(t-1)\left(1 - e^{-\frac{\Delta t}{R_{ij}^{P1}(t-1)C_{ij}^{P1}(t-1)}}\right) I_{ij}(t-1) \tag{4}$$

and

$$V_{ij}^{P2}(t) = e^{-\frac{\Delta t}{R_{ij}^{P2}(t-1)C_{ij}^{P2}(t-1)}} V_{ij}^{P2}(t-1) + R_{ij}^{P2}(t-1)\left(1 - e^{-\frac{\Delta t}{R_{ij}^{P2}(t-1)C_{ij}^{P2}(t-1)}}\right) I_{ij}(t-1). \tag{5}$$

Based on measurement data (terminal voltage, current) and EIS parameters, the BMS estimates the SOC and SOH of each cell in order to schedule switches in the battery pack. Details are explained in the following subsection.

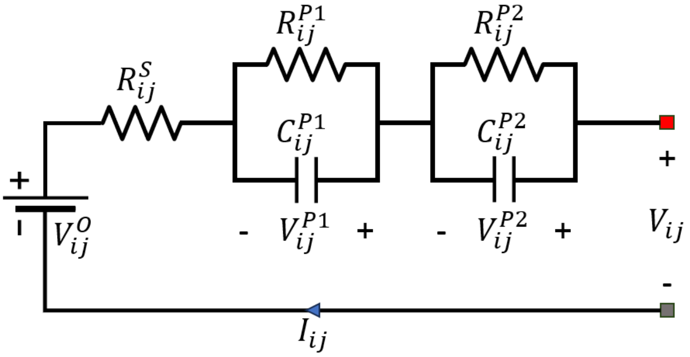


Figure 4. Battery cell model.

2.2. Battery Management System

The BMS schedules the switches in the battery pack and external systems (i.e., the power supply or load demand) according to SOC and SOH estimation, as shown in Figure 3. To estimate SOC and SOH, the BMS monitors the voltage, current, and temperature of each battery cell. We define the SOC of cell (i, j) at time t as the level of charge at time t relative to the maximum battery capacity by [18]

$$SOC_{ij}(t) = SOC_{ij}(t-1) - \frac{\eta \Delta t I_{ij}(t-1)}{M_{ij}(t)}, \tag{6}$$

where $M_{ij}(t)$ is the estimated capacity level of cell (i, j) at time slot t, and η is the Coulombic efficiencies of the discharging and charging processes. We set the measured current $I_{ij}(t-1)$ to positive when discharged and negative when charged, to simplify Equation (6).

Module i, in which cells are connected in parallel, ensures that all cells share the same voltage while their individual currents add up, resulting in a cumulative increase in the total capacity of the module [27]. Therefore, capacity $M_i(t)$ of module i in parallel connection is the sum of the capacities of the individual cells based on Kirchhoff's law as [28]

$$M_i(t) = \sum_{j=1}^{n} M_{ij}(t). \tag{7}$$

We define the SOC of module i at time slot t as the ratio of the total charge level of all cells in module i to the total capacity $M_i(t)$ of module i by [18]

$$SOC_i(t) = \sum_{j=1}^{n} SOC_{ij}(t) \frac{M_{ij}(t)}{M_i(t)}. \tag{8}$$

The SOH of cell (i, j) at time t, which is the ratio of the maximum battery capacity at time t to its rated capacity [29], is defined as

$$SOH_{ij}(t) = \frac{M_{ij}(t)}{M_{ij}^{new}} = \frac{M_{ij}(t)}{M^{new}}, \tag{9}$$

where M_{ij}^{new} is the initial capacity of new cell (i, j); in this paper, we consider all battery cells to be the same type and have the same initial capacity as $M_{ij}^{new} = M^{new}$ [30,31]. The BESS exhibits heterogeneous SOHs between individual cells. From Equation (7), capacity $M_i(t)$ of module i at time slot t is the capacity summation of parallel-connected cells in the given module. Therefore, combining Equation (9), the SOH of module i at time slot t, $SOH_i(t)$, is defined as the SOH average of all parallel-connected cells in module i by

$$SOH_i(t) = \frac{M_i(t)}{nM^{new}} = \frac{1}{n} \sum_{j=1}^{n} SOH_{ij}(t), \tag{10}$$

where $SOH_{ij}(t)$ is the SOH of cell j in module i at time slot t.

The SOH of individual aging cells is not uniform, leading to SOH inconsistencies between modules. The battery pack, in which modules are connected in series, decreases the terminal voltage, limiting the fulfillment of demand with lower SOH modules [32]. Therefore, we define the SOH of the battery pack, $SOH_{\mathcal{P}}(t)$, as the lowest SOH of series-connected modules by [30,32]

$$SOH_{\mathcal{P}}(t) = \min_i SOH_i(t), \tag{11}$$

where $SOH_i(t)$ is the SOH of module i at time slot t. The SOH of the battery pack $SOH_{\mathcal{P}}(t)$, which represents battery pack aging, is a non-increasing function until the end of its life cycle. In this paper, we define the lifetime of the battery pack, T_{EoL}, as the first time slot after the battery pack $SOH_{\mathcal{P}}(t)$ reaches a threshold, denoted as Th_{EoL}, which is assumed to be 60% of SOH as seen in Figure 1, by

$$T_{EoL} = \arg\min_{t} \; \{SOH_{\mathcal{P}}(t) \leq Th_{EoL}\}. \tag{12}$$

The BMS protects battery cells from overcharge, overdischarge, overheating, and excess current by controlling the BESS to fulfill demand when discharging to the load, then recharging from the power supply to recover power based on the SOC and SOH information. Note that the load and power supply change continuously under actual conditions, where $d_D(t)$ and $d_C(t)$ represent the power demand and available power supply, respectively, at time slot t and are constrained as [16]

$$d_D(t)d_C(t) = 0 \text{ with } d_D(t), d_C(t) \geq 0, \tag{13}$$

which means the battery pack only charges, discharges, or is idle at the given time slot. We define the first time slot of a discharge process t_D if $d_D(t_D) > 0$ and $d_D(t_D - 1) = 0$. Similarly, the first time slot of a charge process, t_C, is defined if $d_C(t_C) > 0$ and $d_C(t_C - 1) = 0$. When discharging, the BMS considers the amount of discharged power, $l_D(t)$, and the maximum dischargeable power, $E_D(t)$, of the battery pack at time slot t as [33]

$$l_D(t) = \sum_{i=1}^{m} \big(SOC_i(t_D) - SOC_i(t)\big) M_i(t) \tag{14}$$

and

$$E_D(t) = \sum_{i=1}^{m} \big(SOC_i(t_D) - SOC_{min}\big) M_i(t), \tag{15}$$

where SOC_{min} indicates the lower bound of the SOC to prevent overdischarge, and t_D is the first time slot of the discharge process containing t. When charging, the BMS considers the amount of charged power, $l_C(t)$, and the maximum chargeable power, $E_C(t)$, of the battery pack at time slot t as [33]

$$l_C(t) = \sum_{i=1}^{m} \big(SOC_i(t) - SOC_i(t_C)\big) M_i(t) \tag{16}$$

and

$$E_C(t) = \sum_{i=1}^{m} \big(SOC_{max} - SOC_i(t_C)\big) M_i(t), \tag{17}$$

where SOC_{max} indicates the upper bound of the SOC to prevent overcharging, and t_C is the first time slot of the charge process containing t.

The BMS controls the battery pack discharge if $d_D(t) > 0$ and $l_D(t) < E_D(t)$, or charge if $d_C(t) > 0$ and $l_C(t) < E_C(t)$. Otherwise, the battery pack returns to idle for the remainder of the discharge or charge process. To that end, after estimating SOCs and SOHs, and controlling the charging or discharging process of the battery pack, the BMS schedules all switches in the battery pack in order to extend the lifetime and to reduce the SOH imbalance among the cells and modules.

3. Problem Formulation

The aim of this paper is to extend the lifetime of the battery pack in the BESS. To maximize the battery pack lifetime, we formulate the lifetime maximization of the battery pack as

$$\begin{aligned} \max \quad & T_{EoL} \\ \text{s.t.} \quad & I^{-}_{min} \leq I_{ij}(t) \leq I^{+}_{max}, \\ & SOC_{min} \leq SOC_{ij}(t) \leq SOC_{max}, \\ & l_D(t) \leq E_D(t), \\ & l_C(t) \leq E_C(t), \end{aligned} \tag{18}$$

where I^{+}_{max} and I^{-}_{min} represent the discharge current and charge current thresholds to limit the discharging and charging rates, respectively; SOC_{min} and SOC_{max} represent the lower and upper limits of the SOC, respectively, which are necessary to prevent overdischarging and overcharging; $l_D(t)$ and $l_C(t)$ represent the discharged and charged power, respectively, until time slot t; $E_D(t)$ and $E_C(t)$, respectively, indicate the maximum power load of the battery pack when discharging and charging. Maximizing battery pack lifetime T_{EoL} means maximizing the number of time slots in this second life cycle (Stage 2 in Figure 1). To carry this out, we minimize the rate of battery pack aging (i.e., the SOH reduction in the battery pack). We first define the SOH reduction in the battery pack at time slot t as

$$\Delta SOH_{\mathcal{P}}(t) = SOH_{\mathcal{P}}(t-1) - SOH_{\mathcal{P}}(t), \tag{19}$$

where $SOH_{\mathcal{P}}(t-1)$ and $SOH_{\mathcal{P}}(t)$ denote the SOH of the battery pack, respectively, at time slots $t-1$ and t. $SOH_{\mathcal{P}}(t)$ is a non-increasing function, so we constrain it with

$$\Delta SOH_{\mathcal{P}}(t) \geq 0. \tag{20}$$

To achieve this, the problem is framed to minimize the reduction in SOH of the battery pack, which is mathematically expressed as

$$\begin{aligned} \min \quad & \sum_t \Delta SOH_{\mathcal{P}}(t) \\ \text{s.t.} \quad & \Delta SOH_{\mathcal{P}}(t) \geq 0, \\ & I^{-}_{min} \leq I_{ij}(t) \leq I^{+}_{max}, \\ & SOC_{min} \leq SOC_{ij}(t) \leq SOC_{max}, \\ & l_D(t) \leq E_D(t), \\ & l_C(t) \leq E_C(t). \end{aligned} \tag{21}$$

4. The Proposed Algorithm

To enhance the lifetime of the battery pack in the BESS, we propose a battery management algorithm to minimize SOH reduction in the battery pack. The overall flow of the proposed algorithm is shown in Algorithm 1. At each time slot, the algorithm first gathers measurement data, including the terminal voltage, current, and temperature of each cell. It then estimates the SOC and SOH (Algorithm 2) and manages the BESS charging or discharging process based on the dynamic power demand (Algorithm 3). After updating the states in the BESS, the proposed algorithm schedules all switches in the battery pack to prolong its second lifetime based on the cooperative multi-agent deep Q network (Algorithm 4). The following subsections provide a detailed discussion of each component of the proposed algorithm.

Algorithm 1 BESS Scheduling Algorithm

```
Gather measurement data V, I, T
Estimate SOCs and SOHs (Algorithm 2)
Control discharge, charge, and idle processes of the BESS (Algorithm 3)
Update the BESS states
Schedule switches in the battery pack (Algorithm 4)
```

Algorithm 2 EKF-based SOC and SOH estimation

```
Input: Measurement data V, I, T; Data tables
Output: $SOC_{ij}(t)$, $SOH_{ij}(t)$
for each cell $(i, j)$ do
    Estimate state vector $\widehat{x_{ij}}(t)$ and error covariance $\widehat{P_{ij}}(t)$ using (23) and (24)
    Estimate terminal voltage $\widehat{V_{ij}}(t)$ using (28)
    Compute Kalman gain $G_{ij}(t)$ using (32)
    Correct state vector $x_{ij}(t)$ and error covariance $P_{ij}(t)$ using (33) and (34)
    Update $SOC_{ij}(t)$ and $M_{ij}(t)$
    Update $SOH_{ij}(t)$ using (35)
end for
Update SOC and SOH of each module using (8) and (10)
Update SOH of the battery pack using (11)
```

Algorithm 3 Charge, Discharge, and Idle Controlling

```
Input: $l_D(t)$, $l_C(t)$, $d_D(t)$, $d_C(t)$, $E_D(t)$, $E_C(t)$
Output: Discharge, charge, or idle
Determine the current process (discharge, charge, or idle)
if $d_D(t) > 0$ then                                          ▷ Discharging
    if $l_D(t) \geq d_D(t)$ then
        Convert discharge to charge
    else
        if $l_D(t) \geq E_D(t)$ then
            Convert discharge to idle
        else
            Continue to discharge
        end if
    end if
else if $d_C(t) > 0$ then                                     ▷ Charging
    if $l_C(t) \geq d_C(t)$ then
        Convert charge to discharge
    else
        if $l_C(t) \geq E_C(t)$ then
            Convert charge to idle
        else
            Continue to charge
        end if
    end if
end if
```

Algorithm 4 Switch Scheduling

```
Input: state vector $s(t)$
Output: Optimal joint action $a(t)$
Construct main network $\mathcal{Q}_i$ and target network $\overline{\mathcal{Q}}_i$ for each agent
Initialize acquired knowledge $\mathcal{K}$
Select module switch action $a_{m+1}(t)$ by $\epsilon$-greedy policy    ▷ Switch ON/OFF modules
for each agent $i = 1$ to $m$ do
    if module $i$ is OFF then
        Limit action $a_i(t)$    ▷ Switch OFF all cells in module $i$
    else
        Select action $a_i(t)$ by $\epsilon$-greedy policy    ▷ Switch ON/OFF cells in module $i$
    end if
end for
Execute joint action $a(t)$
for each agent $i = 1$ to $m$ do
    Compute local reward $r_i(t)$
end for
Compute global reward $r_{m+1}(t)$
Update new state $s(t+1)$
Add $\langle s_i(t), a_i(t), r_i(t), s_i(t+1) \rangle$ into $\mathcal{K}$ for each agent
Compute target action value $\overline{Q_i}(t)$ for each agent
Perform a gradient descent to minimize loss function $\mathcal{L}(\phi_i(t))$ for each agent
```

4.1. SOC and SOH Estimation

To observe the states of the battery pack, the proposed algorithm estimates the SOC and SOH of each battery cell by gathering its terminal voltage, current, and temperature. To carry this out, the algorithm first uses a fourth-order extended Kalman filter (EKF) to estimate the SOC and SOH of each battery cell, and then updates the SOC and SOH information of each battery module and the battery pack.

For each cell (i, j), we define the corrected state vector of cell (i, j) at time t_k, denoted as $x_{ij}(t_k)$, by

$$x_{ij}(t_k) = [SOC_{ij}(t_k),\ V_{ij}^{P1}(t_k),\ V_{ij}^{P2}(t_k),\ 1/M_{ij}(t_k)]^T. \tag{22}$$

Based on measured current $I_{ij}(t-1)$ and the modeled EIS parameters at the previous time slot, the algorithm estimates state vector $\widehat{x_{ij}}(t)$ and error covariance $\widehat{P_{ij}}(t)$ as

$$\widehat{x_{ij}}(t) = A_{ij}(t-1)x_{ij}(t-1) + B_{ij}(t-1)I_{ij}(t-1) \tag{23}$$

and

$$\widehat{P_{ij}}(t) = A_{ij}(t-1)P_{ij}(t-1)A_{ij}(t-1)^T, \tag{24}$$

where $x_{ij}(t-1)$ is the correct state vector of cell (i, j) at time $t-1$; $A_{ij}(t-1)$ and $B_{ij}(t-1)$ denote the transition matrix and the input matrix, respectively. Matrices $A_{ij}(t-1)$ and $B_{ij}(t-1)$ are defined as

$$A_{ij}(t-1) = \begin{bmatrix} 1 & 0 & 0 & -\eta \Delta t I_{ij}(t-1) \\ 0 & e^{-\frac{\Delta t}{R_{ij}^{P1}(t-1)C_{ij}^{P1}(t-1)}} & 0 & 0 \\ 0 & 0 & e^{-\frac{\Delta t}{R_{ij}^{P1}(t-1)C_{ij}^{P1}(t-1)}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{25}$$

and

$$B_{ij}(t-1) = \begin{bmatrix} 0 \\ R_{ij}^{P1}(t-1)(1-e^{-\frac{\Delta t}{R_{ij}^{P1}(t-1)C_{ij}^{P1}(t-1)}}) \\ R_{ij}^{P2}(t-1)(1-e^{-\frac{\Delta t}{R_{ij}^{P2}(t-1)C_{ij}^{P2}(t-1)}}) \\ 0 \end{bmatrix}, \tag{26}$$

where $I_{ij}(t-1)$ is the measured current of cell (i,j) at $t-1$. EIS parameters R_{ij}^{S}, R_{ij}^{P1}, C_{ij}^{P1}, R_{ij}^{P2}, and C_{ij}^{P2} are functions of SOC_{ij} and SOH_{ij}. Specifically, the EIS parameters are exponential functions of $SOC_{ij}(t-1)$, such as

$$x_1 \exp\big(x_2 SOC_{ij}(t-1)\big) + x_3, \tag{27}$$

where x_1, x_2, and x_3 are real numbers, depending on the SOH of each cell SOH_{ij}. A dataset [34] is used to construct look-up tables of each EIS parameter $\left(R_{ij}^{S}, R_{ij}^{P1}, C_{ij}^{P1}, R_{ij}^{P2}\right.$, or $\left.C_{ij}^{P2}\right)$ based on the above exponential functions. The algorithm estimates terminal voltage $\widehat{V_{ij}}(t)$ using state vector $\widehat{x_{ij}}(t)$ and Jacobian matrices $C_{ij}(t)$ and $D_{ij}(t)$ as

$$\widehat{V_{ij}}(t) = C_{ij}(t)\widehat{x_{ij}}(t) + D_{ij}(t)I_{ij}(t), \tag{28}$$

where $I_{ij}(t)$ is the measured current of cell (i,j) at time slot t. Matrices $C_{ij}(t)$ and $D_{ij}(t)$ are, respectively, defined as

$$C_{ij}(t) = \begin{bmatrix} \frac{\delta V_{ij}^{O}(t)}{\delta SOC_{ij}(t)} & -1 & -1 & 0 \end{bmatrix} \tag{29}$$

and

$$D_{ij}(t) = -R_{ij}^{S}(t). \tag{30}$$

Open-circuit voltage $V_{ij}^{O}(t)$ is defined as the ath-order polynomial function of $SOC_{ij}(t)$ by

$$V_{ij}^{O}(t) = \sum_{b=0}^{a} y_b \big(SOC_{ij}(t)\big)^b, \tag{31}$$

where y_b is a real number depending on SOH_{ij}. The algorithm calculates Kalman gain $G_{ij}(t)$ to consider the error of the estimated value as

$$G_{ij}(t) = \widehat{P_{ij}}(t)C_{ij}(t)^T\left(C_i(t)\widehat{P_{ij}}(t)C_{ij}(t)^T\right)^{-1}. \tag{32}$$

Using the measured terminal voltage of cell (i,j), $V_{ij}(t)$, the algorithm corrects state vector $x_{ij}(t)$ and error covariance $P_{ij}(t)$ as

$$x_{ij}(t) = \widehat{x_{ij}}(t) + G_{ij}(t)\left(V_{ij}(t) - \widehat{V_{ij}}(t)\right) \tag{33}$$

and

$$P_{ij}(t) = \left(1 - G_{ij}(t)C_{ij}(t)\right)\widehat{P_{ij}}(t). \tag{34}$$

From corrected state vector $x_{ij}(t)$, $SOC_{ij}(t)$ and maximum capacity $M_{ij}(t)$ of cell (i,j) at time slot t are updated. The estimation algorithm updates the SOH of cell (i,j) at time slot t by averaging it after completing a charge or a discharge process in the battery pack, as the SOH does not degrade immediately after single or multiple time slots [35], as

$$SOH_{ij}(t) = \begin{cases} \frac{1}{t-t_D+1}\sum_{\tau=t_D}^{t} SOH_{ij}(\tau) & \text{if completing discharge at time slot } t; \\ \frac{1}{t-t_C+1}\sum_{\tau=t_C}^{t} SOH_{ij}(\tau) & \text{if completing charge at time slot } t; \\ SOH_{ij}(t-1) & \text{otherwise} \end{cases}, \quad (35)$$

where t_D and t_C are the first time slots of the discharging and charging processes containing t. To that end, the estimation algorithm updates the SOC and SOH of modules $\left(SOC_i(t), SOH_i(t) \mid \forall i = \{1, 2, \ldots, m\}\right)$ using Equations (8) and (10). $SOH_{\mathcal{P}}(t)$ of the battery pack is also updated using Equation (11). Algorithm 2 summarizes the EKF-based combined SOC and SOH estimation.

4.2. Charge, Discharge, and Idle Controlling

The proposed algorithm controls the charging and discharging processes of the battery pack to prevent overcharging and overdischarging in the BESS. By comparing loaded power $(l_D(t)$, and $l_C(t))$, maximum power load $\left(E_D(t)\text{, and } E_C(t)\right)$, power demand $\left(d_D(t)\right)$, and available power supply $\left(d_C(t)\right)$, the algorithm determines whether the status of the BESS is discharging, charging, or idle in order to control it accordingly.

On the one hand, if the battery pack is discharging, i.e., $d_D(t) > 0$, we calculate the amount of power discharged $l_D(t)$ and maximum dischargeable power $E_D(t)$ using Equations (14) and (15). If $l_D(t)$ reaches power demand $d_D(t)$, the algorithm converts the BESS status from discharging to charging. When $l_D(t)$ reaches the maximum power load of the battery pack when discharging $E_D(t)$, the algorithm lets the battery pack go idle. Otherwise, the discharge process continues.

On the other hand, if the battery pack is charging, i.e., $d_C(t) > 0$, we calculate the amount of power charged $l_C(t)$ and maximum chargeable power $E_C(t)$ by using Equations (16) and (17). If $l_C(t)$ reaches $d_C(t)$, the algorithm converts the BESS status from charging to discharging. When $l_C(t)$ reaches the maximum power load of the battery pack at time slot t when charging $E_C(t)$, the algorithm lets the battery pack go idle. Otherwise, it continues charging. The process of the battery pack charging and discharging controlling is summarized in Algorithm 3.

4.3. Cooperative Multi-Agent Deep Reinforcement Learning-Based Battery Switch Scheduling

A cooperative multi-agent deep Q network (CM-DQN) scheduling algorithm is proposed to control switches in order to minimize the SOH reduction in the battery pack. The algorithm uses $m+1$ agents in which agent $m+1$ collectively evaluates the battery pack and comes up with module switch scheduling for serial connections, while m agents (1 to m) correspond to m modules for scheduling cell switches in parallel networks. While agents 1 to m optimize the local reward (i.e., minimize the SOH reduction in each module) by sharing policies, agent $m+1$ obtains global states (from modules and the BESS) to minimize the SOH reduction in the battery pack in the second life cycle. The proposed algorithm perceives environment state $s(t) = \{s_i(t) \mid i = 1, \ldots, m+1\}$ based on the estimation algorithm (Algorithm 2) and the charge and discharge control algorithm (Algorithm 3), and chooses a joint action $a(t) = \{a_i(t) \mid i = 1, \ldots, m+1\}$ by achieving a cumulative reward $r(t) = \{r_i(t) \mid i = 1, \ldots, m+1\}$. The proposed switch scheduling algorithm first monitors the current environmental state of the battery pack and then derives state vectors for each agent, respectively, as

$$s_i(t) = \begin{cases} \left[\mathcal{C}_i(t),\ \mathcal{H}_i(t),\ I_{\mathcal{P}}(t),\ d_D(t),\ d_C(t)\right], \text{with } i = 1, 2, \ldots, m \\ \left[\mathcal{C}(t), \mathcal{H}(t),\ V_{\mathcal{P}}(t),\ d_D(t),\ d_C(t)\right], \text{with } i = m+1, \end{cases} \quad (36)$$

where $\mathcal{C}(t)$ and $\mathcal{H}(t)$ represent sets of the SOCs and SOHs of m series-connected modules, respectively; $\mathcal{C}_i(t)$ and $\mathcal{H}_i(t)$ are value sets of the SOCs and SOHs of n parallel-connected cells in module i, respectively; $V_{\mathcal{P}}(t)$ and $I_{\mathcal{P}}(t)$ represent the measured terminal voltage and the measured current of the battery pack at time slot t; and $d_D(t)$ and $d_C(t)$ are the load demand and available power supply at time slot t, respectively. Action $a_i(t)$ with $i = 1, \ldots, m+1$ controls switches in the battery pack and is defined as

$$a_i(t) = \begin{cases} \left[\mathcal{X}_{i1}(t), \ldots,\ \mathcal{X}_{in}(t)\right], \text{with } i = 1, 2, \ldots, m \text{ for cell switches} \\ \left[\mathcal{X}_1(t), \ldots,\ \mathcal{X}_m(t)\right], \text{with } i = m+1 \text{ for module switches.} \end{cases} \tag{37}$$

All agents continuously interact with each other by sharing states and policies so the training is synchronous and accurate.

To minimize SOH reduction in the battery pack in accordance with the obtained state, the algorithm utilizes acquired knowledge $\mathcal{K}$ that includes a replay buffer to store in the form of $\langle s_i(t), a_i(t), r_i(t), s_i(t+1) \rangle$ with an experience in each time slot. For each agent $i = \{1, \ldots, m+1\}$, the algorithm constructs a main network $\mathcal{Q}_i$, and a target network $\overline{\mathcal{Q}}_i$ with the same structure and random weights $\phi_i = \overline{\phi}_i$, where $\mathcal{Q}_i$ approximates the action-value function $Q_i\big(s_i(t), a_i(t) \mid \phi_i\big)$ at time slot t, and $\overline{\mathcal{Q}}_i$ computes the target action value $\overline{Q}_i(t)$.

To utilize the past experiences, the proposed algorithm looks at acquired knowledge $\mathcal{K}$ to determine whether state $s_{m+1}(t)$ is in $\mathcal{K}$ or not. If state $s_{m+1}(t)$ belongs to $\mathcal{K}$, the algorithm chooses action $a_{m+1}(t)$ using an ϵ-greedy policy [36]. Specifically, it either chooses a random action with probability $p = \epsilon$ or opts for the action with the highest $Q_{m+1}(t_k)$ with probability $p = 1 - \epsilon$, by

$$a_{m+1}(t) = \begin{cases} \text{random action, with } p = \epsilon \\ \arg\max_{a_{m+1}(t)} Q_{m+1}(t_k), \text{with } p = 1 - \epsilon \end{cases}. \tag{38}$$

In the case in which state $s_{m+1}(t)$ is not in $\mathcal{K}$, scheduling action $a_{m+1}(t)$ is performed at random. We determine which modules are bypassed at time slot t, and then limit scheduling action in each agent of cell switch scheduling (agent 1 to m). If module i is bypassed, we limit action $a_i(t)$ by turning OFF all cells in module i. Otherwise, the algorithm continues to assess the acquired knowledge $\mathcal{K}$ to determine whether state $s_i(t)$ is included in $\mathcal{K}$. If state $s_i(t)$ is in $\mathcal{K}$, the algorithm selects action $a_i(t)$ according to an ϵ-greedy policy as specified in Equation (38). In the case in which state $s_i(t)$ is not in $\mathcal{K}$, scheduling action $a_i(t)$ is chosen randomly. After all actions are chosen, a joint action $a(t) = \{a_i(t) \mid i = 1, \ldots, m+1\}$ is executed.

For agent i of module i corresponding to cell switch scheduling, the algorithm evaluates the local immediate reward as

$$R_i(t) = \mathbb{E}\big[-\Delta SOH_i(t)\big], \tag{39}$$

where $\Delta SOH_i(t)$ is the SOH reduction in module i at time slot t, and is defined as a non-increasing function by

$$\Delta SOH_i(t) = SOH_i(t-1) - SOH_i(t) \geq 0. \tag{40}$$

The algorithm evaluates the local cumulative reward through interactions with the environment and seeks an optimal policy to maximize it as

$$r_i(t) = \sum_{h=t}^{\infty} \gamma^{h-t} R_i(h). \tag{41}$$

To that end, the algorithm computes the global cumulative reward to minimize SOH reduction in the battery pack as

$$r_{m+1}(t) = \sum_{h=t}^{\infty} \gamma^{h-t} R_{m+1}(h), \tag{42}$$

where $R_{m+1}(t_h)$ is the global immediate reward and is computed by

$$R_{m+1}(t) = \mathbb{E}\big[-\Delta\sum_{i=1}^{m} SOH_i(t)\big]. \tag{43}$$

By executing joint action $a(t)$, the algorithm updates new state $s(t+1)$, then stores sample $\langle s_i(t), a_i(t), r_i(t), s_i(t+1)\rangle$ into accquired knowledge $\mathcal{K}$. Target action value $\overline{Q}_i(t)$ is computed by

$$\overline{Q}_i(t) = r_i(t) + \gamma \max_{a_i(t)} Q_i\big(s_i(t+1), a_i(t+1) \mid \overline{\phi}_i\big), \tag{44}$$

where $\gamma \in (0,1]$ is the discount factor that determines the emphasis on future rewards. The CM-DQN updates the acquired knowledge $\mathcal{K}$ by minimizing the loss function $\mathcal{L}\big(\phi_i(t)\big)$ through gradient descent. The loss function is defined as

$$\mathcal{L}\big(\phi_i(t)\big) \leftarrow \mathbb{E}\Big[\Big(\overline{Q}_i(t) - Q_i\big(s_i(t), a_i(t) \mid \phi_i\big)\Big)^2\Big]. \tag{45}$$

Weight $\phi_i(t)$ is updated by the loss function as

$$\phi_i(t) = \phi_i(t-1) + \alpha\nabla\mathcal{L}\big(\phi_i(t)\big) \tag{46}$$

where $\alpha \in (0,1]$ is the learning factor. After computing the loss for an action, the target network $\overline{\mathcal{Q}}_i$ updates its weights to match those of the main network $\mathcal{Q}_i$, i.e., $\overline{\phi}_i = \phi_i$ after P time slots to ensure algorithm stability [36]. Loss function $\mathcal{L}\big(\phi_i(t)\big)$ is minimized so that action value $Q_i\big(s_i(t), a_i(t) \mid \phi_i\big)$ has the same value as target action value $\overline{Q}_i(t)$, which also means that the SOH reduction in the battery pack is minimized. The proposed switch scheduling algorithm is summarized in Algorithm 4. The training process in the CM-DQN is shown in Figure 5.

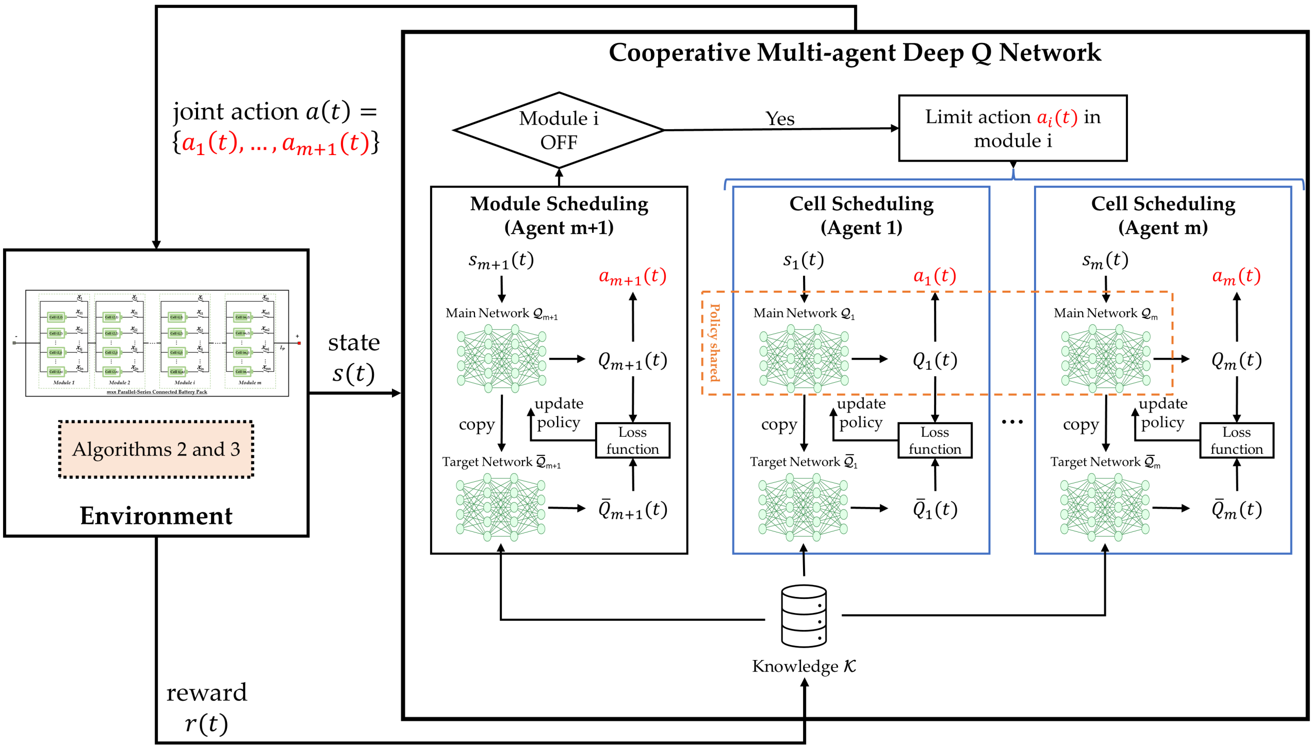

joint action $a(t)$ =
$\{a_1(t), \ldots, a_{m+1}(t)\}$
Cooperative Multi-agent Deep Q Network
Module i OFF
Yes
Limit action $a_i(t)$ in module i
Module Scheduling (Agent m+1)
Cell Scheduling (Agent 1)
Cell Scheduling (Agent m)
$s_{m+1}(t)$
$a_{m+1}(t)$
Main Network $\mathcal{Q}_{m+1}$
$Q_{m+1}(t)$
copy
update policy
Loss function
Target Network $\overline{\mathcal{Q}}_{m+1}$
$\overline{Q}_{m+1}(t)$
$s_1(t)$
$a_1(t)$
Main Network $\mathcal{Q}_1$
$Q_1(t)$
Target Network $\overline{\mathcal{Q}}_1$
$\overline{Q}_1(t)$
Policy shared
$s_m(t)$
$a_m(t)$
Main Network $\mathcal{Q}_m$
$Q_m(t)$
Target Network $\overline{\mathcal{Q}}_m$
$\overline{Q}_m(t)$
…
Knowledge $\mathcal{K}$
state $s(t)$
Algorithms 2 and 3
Environment
reward $r(t)$

Figure 5. The training process in the cooperative multi-agent deep Q network.

5. Performance Evaluation

5.1. Simulation Environment

The BESS was performed via simulation using a lithium-ion battery pack model and was implemented in MATLAB and Simulink 2023b. To assess the performance of the proposed algorithm, we designed a 6 × 4 parallel-series connected battery pack [15], listed in Table 3, including 24 lithium-ion batteries with heterogeneous SOHs. The battery cell was modeled as a second-order Thévenin equivalent battery model with reductions in SOHs by utilizing a NASA dataset [34]. That dataset includes 28 lithium-ion cobalt oxide 18,650 cells with a nominal voltage of 3.7 V and nominal capacity of 2.2 Ah with a maximum current of 4 A, encompassing real-time measurements of terminal voltage, current, cell temperature, discharging capacity, and EIS impedance readings. We identified the EIS parameters, including V_{ij}^{O}, R_{ij}^{S}, R_{ij}^{P1}, C_{ij}^{P1}, R_{ij}^{P2}, and C_{ij}^{P2}, in the 90% to 60% SOH range by using the dataset. A dynamic power demand condition was used to evaluate the effectiveness of the algorithm. The power demand was obtained by generating values from a uniform distribution within 60 Wh to 100 Wh, while the terminal voltage was obtained by generating values from a uniform distribution ranging from 13.75 V to 16.25 V (at least four series-connected modules ON at a given time because the nominal voltage of each cell is 3.7 V). We set the load current for both discharging and charging the battery pack to 8 A (at least two parallel-connected batteries ON at a given time because the maximum output current of one battery is 4 A).

Table 3. Initial SOHs of the cells and modules.

	Module 1	**Module 2**	**Module 3**	**Module 4**	**Module 5**	**Module 6**
$SOH_{i1}(0)$	90.00%	82.17%	79.23%	89.96%	86.72%	82.84%
$SOH_{i2}(0)$	86.77%	84.72%	88.16%	84.23%	83.46%	86.11%
$SOH_{i3}(0)$	84.13%	80.02%	87.35%	85.76%	79.98%	78.72%
$SOH_{i4}(0)$	78.15%	79.28%	81.19%	87.54%	81.75%	80.14%
$SOH_{i}(0)$	84.77%	81.55%	83.98%	86.87%	82.98%	81.95%
$SOH_{\mathcal{P}}(0)$	81.55%					

Two deep Q network architectures were constructed for cell scheduling and module scheduling, each including one input layer, two hidden layers, and one output layer. The number of hidden layers and the number of neurons in each hidden layer can be selected by trial and error [37]. To determine the optimal network size, including the number of hidden layers and the number of neurons in each layer, two approaches are commonly used: the constructive approach and the destructive approach [38]. We used a constructive approach to network sizing [39]. We started with a small network and gradually added neurons or hidden layers to improve the performance of the network, where the number of hidden layer neurons varied among 64, 128, and 256, and the number of hidden layers started from 1. Both deep Q network architectures with two 128-dimension hidden layers had the maximum cumulative reward with minimum episodes. For cell scheduling, the input layer included 11 neurons corresponding to the number of dimensions in state s_i of module i with four parallel-connected cells. The output layer consisted of 12 neurons corresponding to possible actions, a_i, for cell switch scheduling. For module scheduling, the input layer included 15 neurons corresponding to the number of dimensions in state s_{m+1} of the battery pack with six series-connected modules. The output layer consisted of 15 neurons corresponding to possible actions, a_{m+1}, for module switch scheduling. During learning, we set the learning rate α to 0.001, which helped reduce the loss function $\mathcal{L}(\phi_i(t))$ episodes and balance the speed of learning with the stability of the training process. We set the discount factor γ to 0.99 to effectively balance prioritizing long-term cumulative rewards, i.e., minimizing SOH reduction in the battery pack, while avoiding slowing down the convergence process. Other simulation parameters are summarized in Table 4.

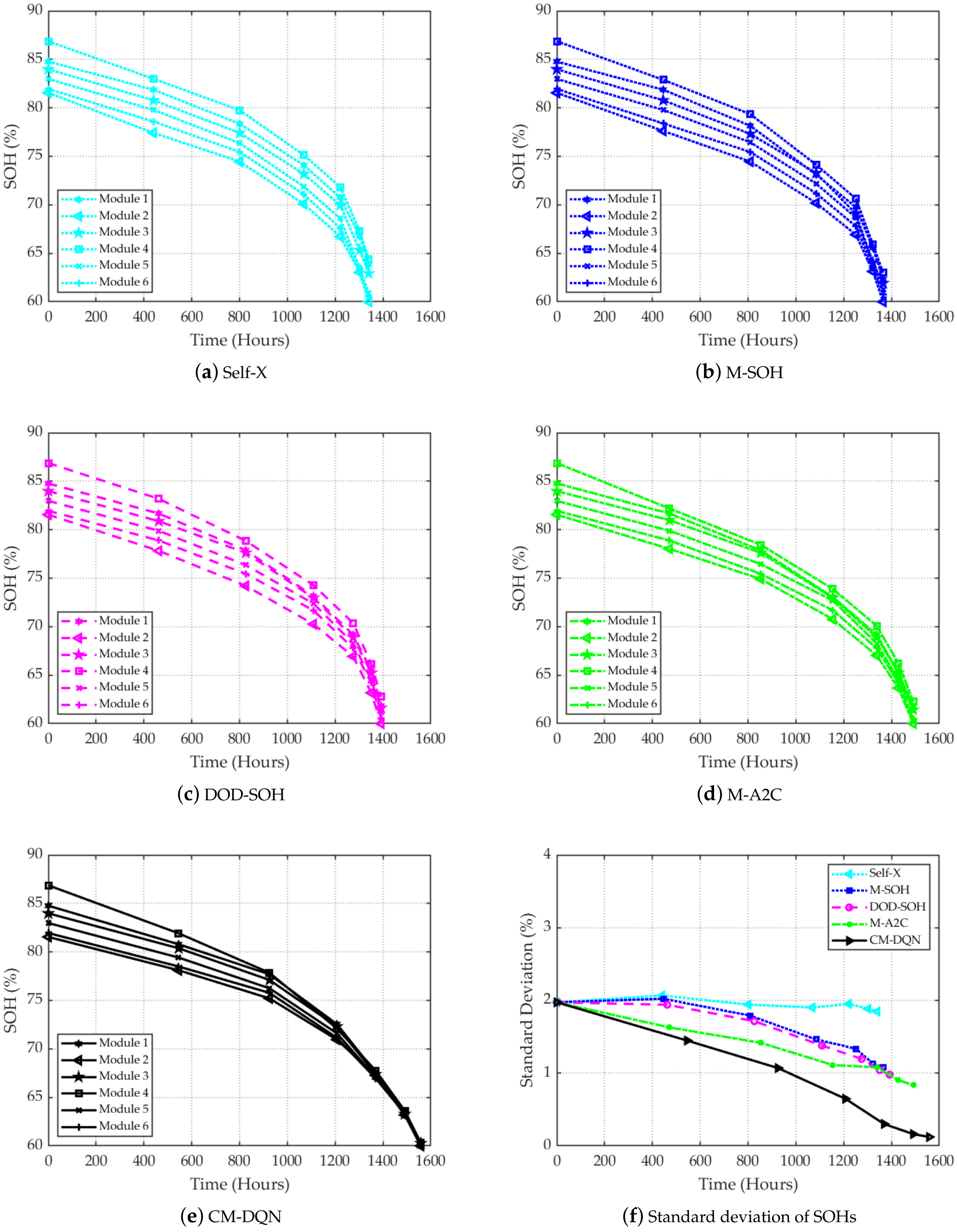
Table 4. Simulation parameters.

Parameter	Value
Number of battery cells	6 × 4
Battery type	Lithium-ion 3.7 V/2.2 Ah
Total capacity (new)	195.36 Wh
$I_{discharge}$	8 A
I_{charge}	−8 A
I^{-}_{min}, I^{+}_{max}	−4 A, 4 A
SOC_{min}, SOC_{max}	10%, 90%
η	1 (discharge)/0.98 (charge)
Total working time	1600 h
Δt	10 min
Hidden layers in each network	2
Neurons in each hidden layer	128
Neurons in input layer (cell scheduling)	11
Neurons in output layer (cell scheduling)	12
Neurons in input layer (module scheduling)	15
Neurons in output layer (module scheduling)	15
Learning rate (α)	0.001
ϵ-greedy value	0.9
Discount factor (γ)	0.99
Period of target network update P	100 time slots

For the performance evaluation, we first investigated the effect of the proposed algorithm on SOH balancing to enhance the lifetime of the battery pack. Then, we evaluated the impact of the proposed algorithm on dynamic power demand. To validate the performance of the proposed cooperative multi-agent deep Q network (denoted as CM-DQN) algorithm, we compare it with previous works, including a self-X multicell battery algorithm [15] denoted as Self-X, a multilayer SOH equalizer [22] denoted as M-SOH, a DOD-SOH balancing algorithm [17] denoted as DOD-SOH, and a multi-actor–critic scheduling algorithm [25] denoted as M-A2C.

5.2. Capacity Balancing and Battery Pack Lifetime

We evaluated the performance of the proposed algorithm on balancing the SOH of modules, as shown in Figure 6. The SOH balancing of modules in the battery pack in all the algorithms is shown in Figure 6a–e. The proposed algorithm (CM-DQN) achieved more SOH balancing than previous work. During learning, the CM-DQN framework determines the optimal actions known as switches to connect or bypass cells and modules by observing BESS states such as the SOCs and SOHs of cells and modules, power demand $d_D(t)$, or available power supply $d_C(t)$. CM-DQN ensures that no single module degrades significantly faster than the others by fully observing the states in the BESS (including SOC, SOH of modules and cells, charging and discharging states, and the power demand). Figure 6f compares the standard deviation in SOHs among the modules over time for each algorithm. CM-DQN exhibited the lowest standard deviation (close to zero), compared to previous work, further affirming its ability to maintain SOH balance among modules. Self-X showed an SOH imbalance that was almost unchanged without considering SOHs. Other methods (M-SOH, DOD-SOH, and M-A2C), which considered SOHs, reduced the SOH imbalance but, without observations of the states in the BESS, the standard deviation was still higher than CM-DQN, indicating that SOHs between modules were not balanced until the end of the lifetime. With heterogeneous SOHs in the battery pack, CM-DQN offered better performance than other algorithms by balancing SOHs among the modules.



(**a**) Self-X

(**b**) M-SOH

(**c**) DOD-SOH

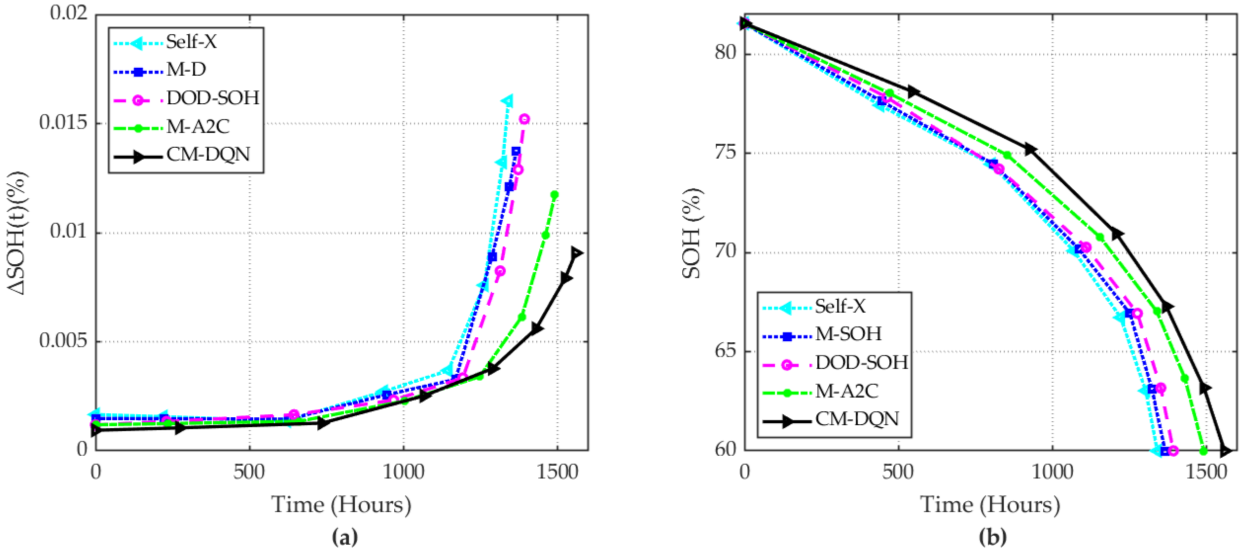
(**d**) M-A2C

(**e**) CM-DQN

(**f**) Standard deviation of SOHs

Figure 6. SOH balancing with different methods.

The performance of the proposed algorithm on battery pack lifetime, measured by minimizing SOH reduction, was evaluated and is shown in Figure 7. By balancing the SOHs among the modules in the battery pack, the proposed algorithm (CM-DQN) achieved the lowest SOH reduction in the battery pack compared to the other algorithms by minimizing the SOH reduction for each time slot, as shown in Figure 7a. The reduction in SOH under Self-X and M-SOH increased rapidly after 1200 h, compared to the other algorithms, due to the lack of consideration of both SOC and SOH. M-A2C had the closest performance to CM-DQN, but ignoring external systems information caused the SOH reduction, $\Delta SOH_{\mathcal{P}}(t)$, to increase faster as the battery lifetime weakened after 1250 h. The SOH reduction in the battery pack was minimized with CM-DQN, resulting in an extended battery pack lifetime, as shown in Figure 7b. The proposed algorithm CM-DQN achieved a longer battery pack lifetime compared to previous work. The SOH of the battery pack under CM-DQN reached 60% (the end of its second life) after a working time of 1558 h. By not observing the states in the BESS, all other work finished the second lifetimes significantly faster than CM-DQN. Table 5 shows that the battery pack lifetime improvement of the proposed algorithm (CM-DQN) compared to previous work was up to 16.27%. Hence, the proposed algorithm can efficiently schedule switches in the battery pack to maximize the BESS lifetime and balance the SOHs among the modules and cells.



(a) **(b)**

Figure 7. Extending lifetime of the battery pack with (**a**) SOH reduction per time slot, and (**b**) maximum lifetime of the battery pack until reaching 60% of SOH.

Table 5. Battery pack lifetime improvement of the proposed algorithm compared with previous work.

Method	Self-X	M-SOH	DOD-SOH	M-A2C
Extended lifetime by CM-DQN	16.27%	14.14%	11.93%	4.49%

To show the impact of SOC balancing on the lifetime of the battery pack, the standard deviations of the SOCs in a battery pack under different algorithms are shown in Figure 8. Self-X, which only focused on SOCs, had better SOC balancing than other algorithms but stopped decreasing after some time and had a worse battery pack lifetime, as shown in Figure 7b. CM-DQN had a continuously decreasing standard deviation over time and it was lower than that of Self-X after 1130 h. Although the proposed algorithm (CM-DQN) focused on only balancing the SOHs of cells by observing SOCs, it also decreased the SOC imbalance among the cells in a battery pack and achieved a longer lifetime. As a result, scheduling cells and modules in a battery pack for SOH balancing is more important to extend the lifetime of the battery pack, as shown in Figures 6f, 7 and 8.

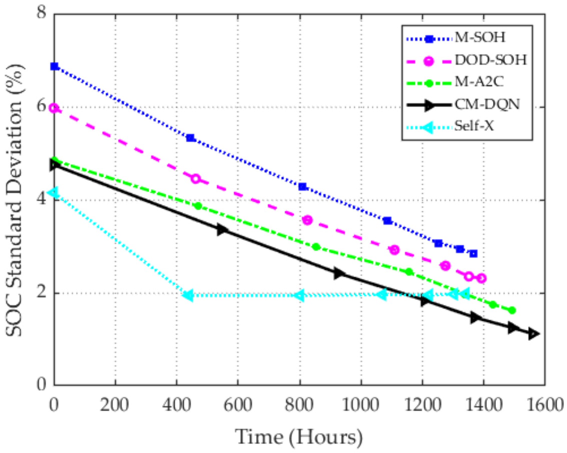


Figure 8. Standard deviation of SOCs among modules.

5.3. Performance of the Proposed Algorithm on Varied Demands

We studied the performance of the proposed algorithm in scenarios with different power demands, as shown in Table 6. We considered various conditions in Scenario 1 with the same mean (80 Wh) and incremental variance (from 8.33 to 533.33), as shown in Figure 9a, and in Scenario 2 with the same variance (133.33) and incremental mean (from 60 Wh to 100 Wh), as shown in Figure 9b. Understanding the impact of demand variance is essential for evaluating the robustness and adaptability of battery management algorithms in real-world applications.

Table 6. Different energy demand conditions.

Scenario	Energy Demand	Mean μ	Variance σ^2
1	U(75 Wh, 85 Wh) *	80 Wh	8.33
	U(70 Wh, 90 Wh)		33.33
	U(60 Wh, 100 Wh)		133.33
	U(50 Wh, 110 Wh)		300
	U(40 Wh, 120 Wh)		533.33
2	U(40 Wh, 80 Wh)	60 Wh	133.33
	U(50 Wh, 90 Wh)	70 Wh	
	U(60 Wh, 100 Wh)	80 Wh	
	U(70 Wh, 110 Wh)	90 Wh	
	U(80 Wh, 120 Wh)	100 Wh	

* U(75 Wh, 85 Wh) was obtained by generating values from a uniform distribution across 75 Wh to 85 Wh.

Figure 10 illustrates the impact of different load demand conditions, characterized by varying demand variance (σ^2), on the lifetime of algorithms under the same mean demand. In Figure 10a, it is observed that, as the demand variance increased from 8.33 to 533.33, the standard deviation for all algorithms also increased. The proposed algorithm minimizes the SOH reduction in the battery pack at each time slot by balancing the SOHs of the battery modules. The charge and discharge control algorithm (Algorithm 3) and observation of dynamic demands $\big($shown in state vectors in (36)$\big)$ helped CM-DQN consistently achieve the lowest standard deviation, indicating robustness against fluctuations in demand. In contrast, the previous algorithms showed an increasing standard deviation proportional to the variation in power demand, indicating a lack of adaptation to demand dynamics. Figure 10b examines the lifetime obtained by the algorithms. The proposed algorithm (CM-DQN) demonstrated a significantly longer lifetime than the others, maintaining a robust performance in SOH balancing and extending lifetime even as demand variance increased. This suggests the proposed algorithm is more adaptive and effective in extending a battery's operational lifetime under varying demand conditions. In summary, the proposed CM-

DQN algorithm outperformed the others in robustness (lower standard deviation and longer lifetime) when subjected to increasing demand variance.

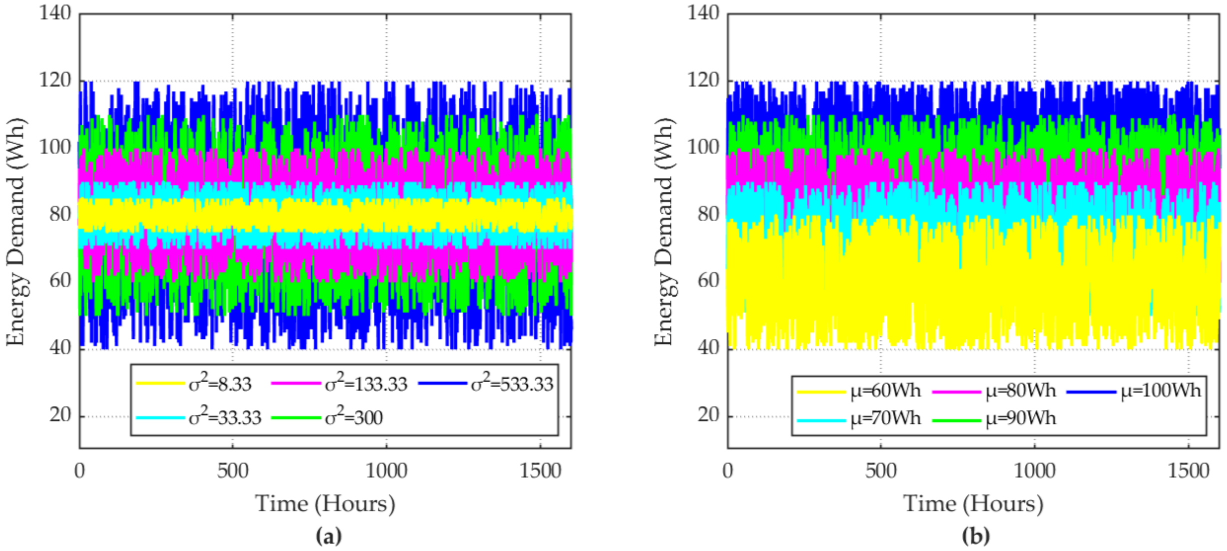


Figure 9. Energy demand profiles with (**a**) Scenario 1, and (**b**) Scenario 2.

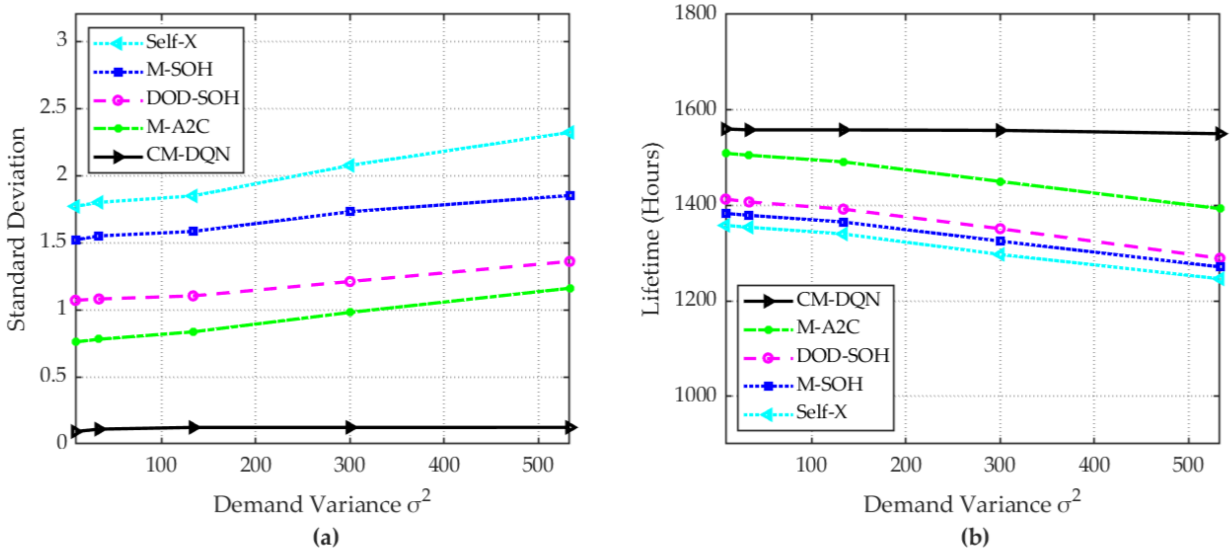


Figure 10. Performance of the scheduling algorithms under dynamic power demand in Scenario 1: (**a**) standard deviations in SOHs among the modules and (**b**) operation times of the battery pack until the SOH reached 60%.

Figure 11 presents the performance of the algorithms under different mean demand conditions (μ) while keeping the variance constant. Figure 11a shows the standard deviation of the algorithms' performance as the demand mean increased from 60 Wh to 100 Wh. The CM-DQN algorithm consistently achieved the lowest standard deviation, indicating its robustness against variations in the demand mean. Figure 11b illustrates the lifetimes of the algorithms as the demand mean increased. The proposed CM-DQN algorithm exhibited the longest lifetime across all mean demand levels, demonstrating its efficiency in extending the battery lifetime under varying power demands. Overall, CM-DQN minimized the SOH reduction in the battery pack in handling different demand mean conditions, providing an extended battery lifetime and an SOH balancing among the modules.

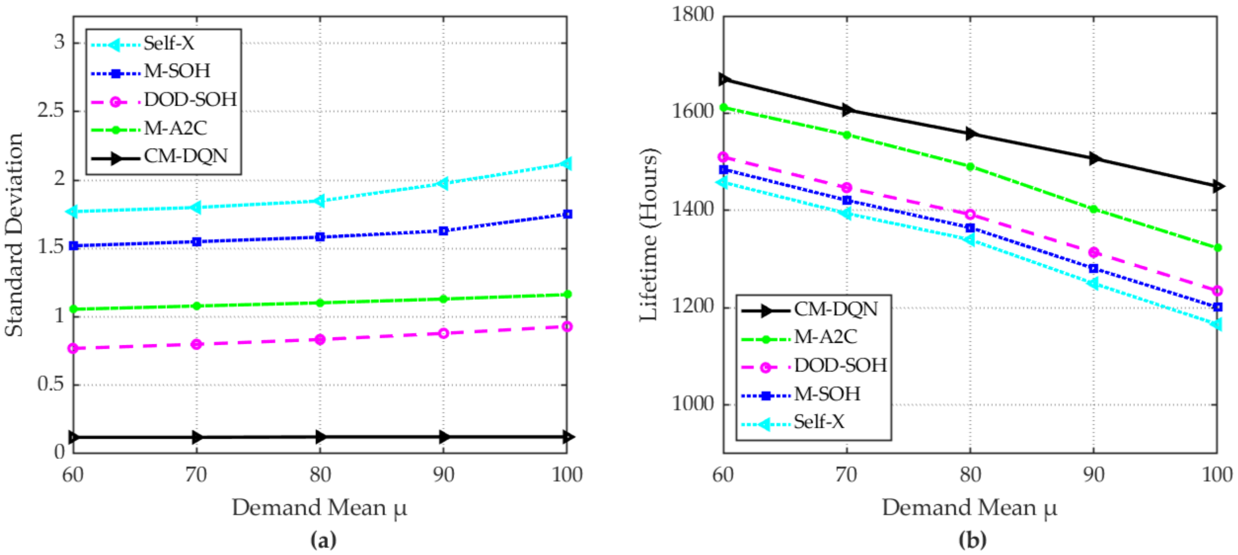


Figure 11. Performance of the scheduling algorithms under dynamic power demand in Scenario 2: (**a**) standard deviation in SOHs among the modules and (**b**) operation time of the battery pack until the SOH reached 60%.

6. Conclusions

In this paper, we proposed a battery management algorithm using a cooperative multi-agent deep Q network to maximize battery lifetimes for the battery pack in a BESS. The BESS consisted of a battery pack in which retired batteries with heterogeneous SOHs are connected in parallel and series. The proposed algorithm scheduled switches in the battery pack by connecting or bypassing battery cells and modules. The algorithm maximized the battery pack lifetime by reducing the SOH imbalance among cells and adapting to varied power demands. Via simulation, we showed that the proposed algorithm outperformed the other algorithms by attaining a more extended battery pack lifetime and reduced SOH imbalance among the cells and modules. The proposed algorithm extended the lifetime of the battery pack by up to 16.27% compared to the other algorithms.

Author Contributions: Conceptualization, N.Q.D., S.M.S. and S.K.; methodology, N.Q.D., S.M.S., S.-J.C. and S.K.; software, N.Q.D.; validation, S.M.S., T.M.D., S.-J.C. and S.K.; formal analysis, N.Q.D., S.M.S., T.M.D. and S.K.; investigation, N.Q.D., S.M.S., T.M.D. and S.K.; writing—original draft preparation, N.Q.D., S.M.S. and T.M.D.; supervision, S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (NRF-2021R1I1A3A04037415) and the Korea Hydro and Nuclear Power Co. (2023).

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: https://www.nasa.gov/intelligent-systems-division/discovery-and-systems-health/pcoe/pcoe-data-set-repository/ (accessed on 22 July 2024).

Conflicts of Interest: The funding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Hunt, G. *USABC Electric Vehicle Battery Test Procedures Manual*; Revision 2; United States Department of Energy: Washington, DC, USA, 1996.
2. Examples for Reuse of Power Batteries. In *Reuse and Recycling of Lithium-Ion Power Batteries*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2017; Chapter 3, pp. 261–334.

3. Canals Casals, L.; García, B.; Aguesse, F.; Iturrondobeitia, A. Second life of electric vehicle batteries: Relation between materials degradation and environmental impact. *Int. J. Life Cycle Assess.* **2015**, *22*, 82–93. [CrossRef]
4. Wang, L.; Yusheng, S.; Wang, X.; Wang, Z.; Zhao, X. Reliability Modeling Method for Lithium-ion Battery Packs Considering the Dependency of Cell Degradations Based on a Regression Model and Copulas. *Materials* **2019**, *12*, 1054. [CrossRef] [PubMed]
5. Martinez-Laserna, E.; Gandiaga, I.; Sarasketa-Zabala, E.; Badeda, J.; Stroe, D.I.; Swierczynski, M.; Goikoetxea, A. Battery second life: Hype, hope or reality? A critical review of the state of the art. *Renew. Sustain. Energy Rev.* **2018**, *93*, 701–718. [CrossRef]
6. Europe's Largest Energy Storage System Now Live at the Johan Cruijff Arena. Available online: https://global.nissannews.com/en/releases/europes-largest-energy-storage-system-now-live-at-the-johan-cruijff-arena/ (accessed on 22 July 2024).
7. Second Life for a Coal Power Plant in Germany. Available online: https://www.electrive.com/2020/11/24/second-life-for-a-coal-power-plant-in-germany/ (accessed on 22 July 2024).
8. Nováková, K.; Pražanová, A.; Stroe, D.I.; Knap, V. Second-Life of Lithium-Ion Batteries from Electric Vehicles: Concept, Aging, Testing, and Applications. *Energies* **2023**, *16*, 2345. [CrossRef]
9. Rouholamini, M.; Wang, C.; Nehrir, H.; Hu, X.; Hu, Z.; Aki, H.; Zhao, B.; Miao, Z.; Strunz, K. A Review of Modeling, Management, and Applications of Grid-Connected Li-Ion Battery Storage Systems. *IEEE Trans. Smart Grid* **2022**, *13*, 4505–4524. [CrossRef]
10. Cactos One Energy Storages. Available online: https://www.cactos.fi/en/product/ (accessed on 22 July 2024).
11. Audi Opens Battery Storage Unit on Berlin EUREF Campus. Available online: https://www.mobilityhouse.com/media/productattachments/files/en20190524_Audi_opens_battery_storage_unit.pdf (accessed on 22 July 2024).
12. A. Market-Intelligence-Report-December-2022. Available online: https://projectcobra.eu/wp-content/uploads/2022/12/Market-Intelligence-Report-December-2022.pdf (accessed on 22 July 2024).
13. Fortum Installs Innovative Battery Solution at Landafors Hydropower Plant in Sweden. Available online: https://www.fortum.com/media/2021/04/fortum-installs-innovative-battery-solution-landafors-hydropower-plant-sweden (accessed on 22 July 2024).
14. Why CellSwitch™ Is Revolutionary for Batteries. Available online: https://www.relectrify.com/technology/cellswitch/ (accessed on 22 July 2024).
15. Kim, T.; Qiao, W.; Qu, L. Power Electronics-Enabled Self-X Multicell Batteries: A Design toward Smart Batteries. *IEEE Trans. Power Electron.* **2012**, *27*, 4723–4733.
16. Doan, N.Q.; Shahid, S.M.; Choi, S.J.; Kwon, S. Deep Reinforcement Learning-Based Battery Management Algorithm for Retired Electric Vehicle Batteries with a Heterogeneous State of Health in BESSs. *Energies* **2024**, *17*, 79. [CrossRef]
17. Yang, X.; Liu, P.; Liu, F.; Liu, Z.; Wang, D.; Zhu, J.; Wei, T. A DOD-SOH balancing control method for dynamic reconfigurable battery systems based on DQN algorithm. *Front. Energy Res.* **2023**, *11*, 1333147. [CrossRef]
18. Ma, Z.; Gao, F.; Gu, X.; Li, N.; Wu, Q.; Wang, X.; Wang, X. Multilayer SOH Equalization Scheme for MMC Battery Energy Storage System. *IEEE Trans. Power Electron.* **2020**, *35*, 13514–13527. [CrossRef]
19. Shen, L.; Li, J.; Meng, L.; Zhu, L.; Shen, H.T. Transfer Learning-Based State of Charge and State of Health Estimation for Li-Ion Batteries: A Review. *IEEE Trans. Transp. Electrif.* **2024**, *10*, 1465–1481. [CrossRef]
20. Chowdhury, S.; Shaheed, M.N.B.; Sozer, Y. State-of-Charge Balancing Control for Modular Battery System with Output DC Bus Regulation. *IEEE Trans. Transp. Electrif.* **2021**, *7*, 2181–2193. [CrossRef]
21. Abdalla, A.A.; Moursi, M.S.E.; El-Fouly, T.H.M.; Hosani, K.H.A. Reliant Monotonic Charging Controllers for Parallel-Connected Battery Storage Units to Reduce PV Power Ramp Rate and Battery Aging. *IEEE Trans. Smart Grid* **2023**, *14*, 4424–4438. [CrossRef]
22. Li, N.; Gao, F.; Hao, T.; Ma, Z.; Zhang, C. SOH Balancing Control Method for the MMC Battery Energy Storage System. *IEEE Trans. Ind. Electron.* **2018**, *65*, 6581–6591. [CrossRef]
23. Wang, X.; Wang, S.; Liang, X.; Zhao, D.; Huang, J.; Xu, X.; Dai, B.; Miao, Q. Deep Reinforcement Learning: A Survey. *IEEE Trans. Neural Networks Learn. Syst.* **2024**, *35*, 5064–5078. [CrossRef] [PubMed]
24. Wu, L.; Lyu, Z.; Huang, Z.; Zhang, C.; Wei, C. Physics-based battery SOC estimation methods: Recent advances and future perspectives. *J. Energy Chem.* **2024**, *89*, 27–40. [CrossRef]
25. Sui, Y.; Song, S. A Multi-Agent Reinforcement Learning Framework for Lithium-ion Battery Scheduling Problems. *Energies* **2020**, *13*, 1982. [CrossRef]
26. Hu, X.; Li, S.; Peng, H. A comparative study of equivalent circuit models for Li-ion batteries. *J. Power Sources* **2012**, *198*, 359–367. [CrossRef]
27. BU-302: Series and Parallel Battery Configurations. Available online: https://batteryuniversity.com/article/bu-302-series-and-parallel-battery-configurations# (accessed on 22 July 2024).
28. von Bülow, F.; Meisen, T. A review on methods for state of health forecasting of lithium-ion batteries applicable in real-world operational conditions. *J. Energy Storage* **2023**, *57*, 105978. [CrossRef]
29. Lipu, M.H.; Hannan, M.; Hussain, A.; Hoque, M.; Ker, P.J.; Saad, M.; Ayob, A. A review of state of health and remaining useful life estimation methods for lithium-ion battery in electric vehicles: Challenges and recommendations. *J. Clean. Prod.* **2018**, *205*, 115–133. [CrossRef]
30. Hua, Y.; Cordoba-Arenas, A.; Warner, N.; Rizzoni, G. A multi time-scale state-of-charge and state-of-health estimation framework using nonlinear predictive filter for lithium-ion battery pack with passive balance control. *J. Power Sources* **2015**, *280*, 293–312. [CrossRef]
31. Chang, C.Y.; Tulpule, P.; Rizzoni, G.; Zhang, W.; Du, X. A probabilistic approach for prognosis of battery pack aging. *J. Power Sources* **2017**, *347*, 57–68. [CrossRef]

32. Jeng, S.L.; Tan, C.M.; Chen, P.C. Statistical distribution of Lithium-ion batteries useful life and its application for battery pack reliability. *J. Energy Storage* **2022**, *51*, 104399. [CrossRef]
33. Lucaferri, V.; Quercio, M.; Laudani, A.; Riganti Fulginei, F. A Review on Battery Model-Based and Data-Driven Methods for Battery Management Systems. *Energies* **2023**, *16*, 7807. [CrossRef]
34. Bole, B.; Kulkarni, C.; Daigle, M. Randomized Battery Usage Data Set. In *NASA Prognostics Data Repository*; NASA Ames Research Center: Moffett Field, CA, USA, 2009.
35. Liu, X.; Li, J.; Yao, Z.; Wang, Z.; Si, R.; Diao, Y. Research on battery SOH estimation algorithm of energy storage frequency modulation system. *Energy Rep.* **2022**, *8*, 217–223. [CrossRef]
36. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; A Bradford Book: Cambridge, MA, USA, 2018.
37. Basheer, I.; Hajmeer, M. Artificial neural networks: Fundamentals, computing, design, and application. *J. Microbiol. Methods* **2000**, *43*, 3–31. [CrossRef]
38. Alpaydin, E. *Introduction to Machine Learning*; MIT Press: Cambridge, MA, USA, 2020.
39. Shahid, S.M.; Ko, S.; Kwon, S. Real-Time Classification of Diesel Marine Engine Loads Using Machine Learning. *Sensors* **2019**, *19*, 3172. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.