Article

Deep Reinforcement Learning-Based Battery Management Algorithm for Retired Electric Vehicle Batteries with a Heterogeneous State of Health in BESSs

Nhat Quang Doan [ID], Syed Maaz Shahid [ID], Sung-Jin Choi [ID] and Sungoh Kwon *[ID]

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.);
sjchoi@ulsan.ac.kr (S.-J.C.)
* Correspondence: sungoh@ulsan.ac.kr

Abstract: In this paper, we propose a battery management algorithm to optimize the lifetimes of retired lithium batteries with heterogeneous states of health in a battery energy storage system under dynamic power demand. A battery energy storage system allows for the use of retired lithium batteries for applications such as backup power in homes, data centers, etc. In a battery energy storage system, a battery pack consists of several retired batteries connected in parallel or in series to fulfill the required power demand. Owing to the retired batteries' different capacity levels, i.e., states of health, a scheduling strategy is required to turn battery cells inside the battery pack on and off such that the secondary lifetimes of the retired batteries are extended. To establish the optimal scheduling policy, it is necessary to determine the correct states of each battery cell, including the state of charge and the state of health. To that end, the proposed algorithm first estimates the state of charge and state of health for all cells based on data measured using an extended Kalman filter. Then, a deep reinforcement learning scheduling algorithm is implemented to connect/disconnect the battery cells to/from the battery pack based on their states. Via simulation, we show that the proposed algorithm estimates the state of charge and state of health of each battery cell with low error and extends the lifetime of battery packs by 20.6%, compared to methods proposed in previous works.

Keywords: battery management system; battery energy storage system; deep reinforcement learning; extended Kalman filter; retired lithium-ion batteries; SOH estimation

1. Introduction

Lithium-ion batteries have become an essential component of modern life, powering everything from smartphones to electric vehicles (EVs) [1]. Their advantages include high energy efficiency, minimal memory effects, extended lifespans, and low self-discharge rates compared to other battery types, and they are now widely used [2]. However, lithium-ion batteries for EVs have a limited lifespan and (eventually, for safety) require replacement when their capacity drops to 80% or lower [3]. The total amount of retired battery-pack power is forecast to reach 120 GWh globally by 2030 [4]. This will create a significant amount of waste and financial burden, particularly as the demand for lithium-ion batteries continues to grow. As a result, it is becoming increasingly urgent and necessary to identify solutions to reuse retired lithium-ion batteries.

One promising application for retired lithium-ion batteries is in battery energy storage systems (BESSs) that can then be used for backup power in homes, EV charging stations, or telecommunication and data center systems [5]. BESSs have the potential to significantly reduce the demand for new batteries and can help reduce the environmental impact of battery production. A battery energy storage system (BESS) has a battery pack in which multiple batteries are connected in parallel or in series to increase the capacity or voltage of the battery pack. A switch is added to each battery cell to connect it to or disconnect

it from the battery pack [6]. Cells in a battery pack have different capacity levels (i.e., heterogeneous states of health), which hinders the effective utilization of the batteries and, consequently, affects the performance of the battery pack. A scheduling policy is required to control the switches in battery cells to prolong the lifetime of the battery pack in the BESS and reduce the imbalanced capacities of battery cells.

For an optimal scheduling policy in a BESS, correct identification of battery characteristics, including the state of charge (SOC) and state of health (SOH), is important. The SOC of a battery is the level of charge relative to the battery's capacity, whereas the SOH is the ratio of the maximum battery charge to its rated capacity. The relationships between SOC and SOH are illustrated in Figure 1. Information on SOC and SOH in a scheduling policy protects the battery cells by preventing them from overcharging or discharging excessively and increases the capacity of the BESS. SOC and SOH parameters cannot be measured directly from a battery cell. Instead, they are estimated through measurable parameters such as voltage, current, and cell temperature. Based on the states of the battery cells in a pack (including the SOC and SOH), the ON/OFF switches for the batteries in the pack are scheduled so that the states of health of all batteries are balanced. As a result, the lifetime of the battery pack is extended. Therefore, the correct estimation of SOC and SOH, along with the scheduling of battery cell switches, is necessary to optimize the performance of a BESS.
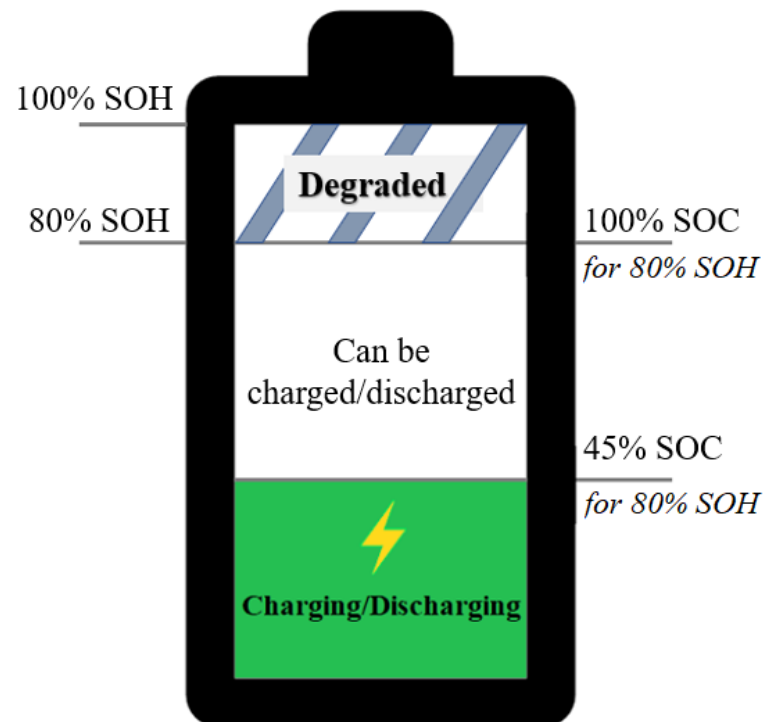

Figure 1. The relationships between state of charge and state of health.

Battery state estimation approaches have been explored in the literature. The Coulomb counting method [7,8] calculates the SOC of cells by counting the amount of charge that enters or exits a cell. However, the Coulomb counting method was unable to measure the SOC of cells in an online parallel-connected battery pack following a sharp drop in SOH. An algorithm was proposed for the online estimation of SOC using deep learning [9], but this algorithm ignores the estimation of SOH. A neural network was used to estimate SOH using experimental datasets [10]. However, the authors did not consider SOC estimation. Information on both SOCs and SOHs of battery cells is required for efficient scheduling of a battery pack to optimize BESS performance. The authors of [11] proposed joint lithium battery SOC and SOH estimation using a data-driven method. This approach required a

large dataset to train the model before operation on site. Kalman filter-based approaches can estimate SOC and SOH levels [12] but depend on the correctness of electrochemical impedance spectroscopy (EIS) parameters, including a resistor and one or more resistor–capacitor (RC) pairs. However, the effect of SOH reduction on EIS parameters is ignored in Kalman filter approaches [13]. In [14], SOH reduction was considered to reidentify EIS parameters, but the SOH was updated only offline. Several researchers have studied the problem of cell scheduling in a parallel-connected battery pack. The authors of [15] utilized a fuzzy logic control strategy to adjust the number of cells in a circuit in accordance with the load demand for the purpose of reducing loop current, which leads to battery inconsistency. In [16], battery resistance degradation was monitored to detect weak cells and disconnect them from the battery pack. This approach solved the issue of mismatched characteristics but requires a complex measuring system or incurs a high computational burden. In [17], the weighted-k round-robin (kRR) scheduling framework was proposed to extend the lifetime of a battery pack by considering load demand and SOH reduction. However, kRR-based scheduling can be implemented only for a fixed model, i.e., the number of cells in the battery pack or the battery models inside the pack cannot change. In [6], a multiactor–critic method was proposed to solve battery scheduling problems. This approach prolonged the lifetime of the battery pack and reduced the imbalance between the batteries but ignored dynamic power demand. In [18], a strategy for a battery management system was proposed, including SOC estimation using an extended Kalman filter algorithm and a scheduler to reduce the difference between the SOCs of battery cells. However, SOH and power demand were not considered in that approach. The main challenge is to determine the accurate state (i.e., SOC and SOH) of a battery cell in a battery pack, then schedule the turning ON/OFF of battery cells based on their current states such that the imbalance in SOHs of cells is reduced.

The main contributions of our work are as follows:

- A scheduling algorithm is proposed to maximize the lifetime of a battery pack consisting of parallel-connected battery cells with heterogeneous states of health in a BESS.
- We define the battery lifetime maximization problem as the reduction in the SOH of a battery pack that can be achieved by reducing the imbalance in the SOHs of battery cells in a battery pack.
- A deep reinforcement learning (DRL) framework is implemented in the scheduling algorithm that uses battery cells' states to set their ON/OFF status and balance the SOHs.
- To measure the battery cells' states to schedule their ON/OFF status, an extended Kalman filter (EKF)-based algorithm is proposed to estimate SOC and SOH.
- A dataset of real measurements is used to determine the accuracy of the proposed estimation algorithm. The proposed algorithm achieves minimal error compared to methods proposed in other works. Simulation results show that the proposed algorithm outperforms previous studies by extending the lifetime of a battery pack under constant and dynamic power demands.

The remainder of this paper is organized as follows. Section 2 discusses the proposed parallel-connected battery model and the scheduling issues. Section 3 presents the framework of the proposed combined algorithm, which includes EKF-based and DRL-based algorithms. Section 4 describes the simulation and presents the results and impacts of the algorithm. Finally, we conclude this work in Section 5.

For ease of presentation, the key notations listed in Table 1 are used throughout this paper.

Table 1. Summary of notations.

Notation	Definition
$\mathcal{W}$	Operational time
$\mathcal{V}(t_k)$	Set of measured cell voltages at time t_k
$\mathcal{I}(t_k)$	Set of measured cell currents at time t_k
$\mathcal{T}(t_k)$	Set of measured cell temperatures at time t_k
$\mathcal{C}(t_k)$	Set of cells' SOC values at time t_k
$\mathcal{H}(t_k)$	Set of cells' SOH values at time t_k
$SOH_\mathcal{P}(t_k)$	SOH of the battery pack at time t_k
$V_i(t_k)$	Measured terminal voltage of cell i at time t_k
$I_i(t_k)$	Measured current of cell i at time t_k
$T_i(t_k)$	Measured cell temperature of cell i at time t_k
$\mathcal{X}_i(t_k)$	ON/OFF switch of cell i
V_{Oi}	Open-circuit voltage of cell i
R_{si}	Internal resistance of cell i
R_{pi}, C_{pi}	Resistor–capacitor pair of cell i
$l_D(t_k^j)$	Discharging power load in cycle j up to time t_k
$l_C(t_k^j)$	Charging power load in cycle j up to time t_k
η	Efficiencies of the discharging/charging process

2. System Model

2.1. Overall System

In this paper, we consider a parallel-connected BESS [19,20] with a power supply and a load, as shown in Figure 2. The BESS comprises a battery pack and a battery management system (BMS) connected to a power supply and a load. We consider a discrete-time model, where the working time ($\mathcal{W}$) is divided into w time slots such that $\mathcal{W} = \{t_k \mid k = 1, 2, ..., w\}$ with durations of $\Delta t = t_k - t_{k-1}$.
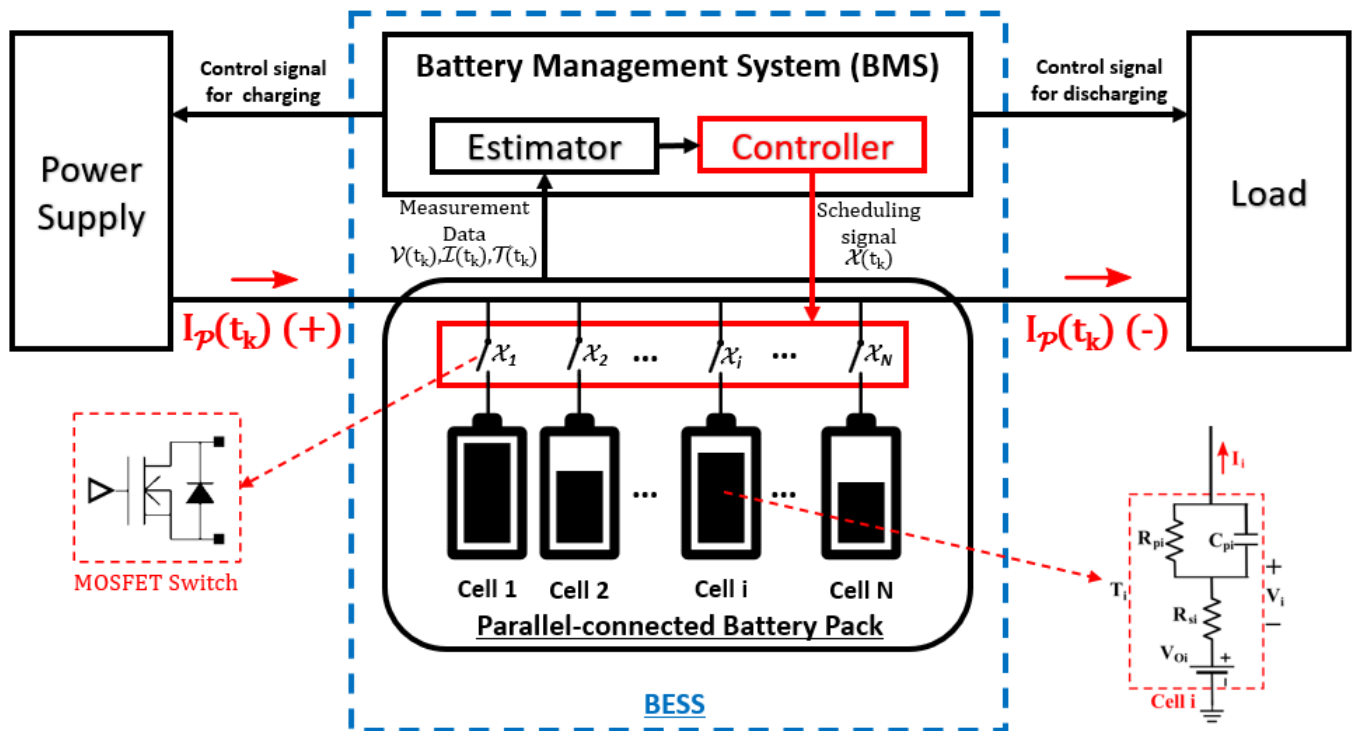

Figure 2. Implementation of a parallel-connected BESS.

The battery pack consists of N lithium battery cells connected in parallel. A first-order Thévenin equivalent model is considered for the cells [21]. Cell $i \in \mathcal{N} = \{1, 2, ..., N\}$ has

EIS parameters including an open-circuit voltage (V_{Oi}); internal resistance (R_{si}); and an RC pair, which includes a resistor (R_{pi}) and capacitor (C_{pi}) connected in parallel. The terminal voltage of cell i at time t ($V_i(t)$) is computed as

$$V_i(t_k) = V_{Oi}(t_k) - V_{pi}(t_k) - R_{si}(t_k)I_i(t_k), \tag{1}$$

where $V_{pi}(t_k)$ is the polarization voltage applied to the parallel RC network, calculated as [14]

$$V_{pi}(t_k) = e^{-\frac{\Delta t}{R_{pi}(t_{k-1})C_{pi}(t_{k-1})}} V_{pi}(t_{k-1}) + R_{pi}(t_{k-1})\left(1 - e^{-\frac{\Delta t}{R_{pi}(t_{k-1})C_{pi}(t_{k-1})}}\right)I_i(t_{k-1}). \tag{2}$$

There are N switches corresponding to N cells linking them to the battery circuit. $\mathcal{X}_i(t_k)$ shows whether a switch of cell i is connected to a battery circuit or not, such that

$$\mathcal{X}_i(t_k) = \begin{cases} 1, & \text{if cell } i \text{ is ON} \\ 0, & \text{if cell } i \text{ is OFF} \end{cases} \tag{3}$$

Similarly, sets $\mathcal{V}(t_k)$, $\mathcal{I}(t_k)$, and $\mathcal{T}(t_k)$ consist of terminal voltages, currents, and temperatures of all cells at time t_k, respectively.

A BMS monitors the states of the battery pack and estimates both the SOC and the SOH of cells in order to schedule the switches in the battery pack. We mathematically define the SOC of cell i at time t_k as

$$SOC_i(t_k) = SOC_i(t_{k-1}) - \frac{\eta \Delta t I_i(t_{k-1})}{M_i(t_k)}, \tag{4}$$

where $M_i(t_k)$ is the capacity level of cell i at time (t_k), and η is the Coulombic efficiencies of the discharging or charging process. Similarly, the SOH of cell i at time t_k is defined as

$$SOH_i(t_k) = \frac{M_i(t_k)}{M_{new}}, \tag{5}$$

where M_{new} is the initial capacity of new cell i. Sets $\mathcal{C}(t_k)$ and $\mathcal{H}(t_k)$ consist of the SOCs and SOHs of all the cells at time t_k, respectively. We define the SOH of the battery pack ($SOH_{\mathcal{P}}(t_k)$) as

$$SOH_{\mathcal{P}}(t_k) = \min_{i \in \mathcal{N}}\left(\mathcal{H}(t_k)\right) \tag{6}$$

Power supply and load are used for charging and discharging of the battery pack. The battery pack current at time t_k ($I_{\mathcal{P}}(t_k)$) has a positive value when discharging and a negative value when charging. The battery pack fulfills the load demand when discharging, then recharges to recover the corresponding amount of power. The process of complete charging and discharging of a battery pack is referred to as a cycle. During the working time ($\mathcal{W}$), an arbitrary cycle (j) has multiple time slots based on the power demand. If time slot t_k belongs to cycle j, we consider $l_D(t_k^j)$ and $l_C(t_k^j)$ to be the amount of power load when discharging and charging in cycle j, respectively, up to time slot t_k, which are calculated as

$$l_D(t_k^j) = \sum_{\tau=\varkappa}^{k} \sum_{\forall i \in \mathcal{N}} \eta V_i(t_\tau) I_i(t_\tau) \Delta t, \tag{7}$$

and

$$l_C(t_k^j) = \sum_{\tau=\varkappa}^{k} \sum_{\forall i \in \mathcal{N}} \eta V_i(t_\tau) I_i(t_\tau) \Delta t. \tag{8}$$

where $\varkappa$ represents the slot number when cycle j starts, i.e., cycle j starts at time $t_\varkappa$.

2.2. Problem Formulation

The objective of this paper is to prolong the lifetime of a battery pack by reducing the rate of aging in cells. To that end, the problem is formulated to minimize the SOH reduction of the battery pack during working time ($\mathcal{W}$), which is mathematically expressed as

$$\min \quad \sum_{k=1}^{w} \Delta SOH_{\mathcal{P}}(t_k)$$

$$\begin{aligned}
\text{s.t.} \quad & \Delta SOH_{\mathcal{P}}(t_k) \geq 0, \\
& I_{min}^- \leq I_i(t_k) \leq I_{max}^+, \\
& SOC_{min} \leq SOC_i(t_k) \leq SOC_{max}, \\
& l_D(t_k^j) \geq d(t_k^j), \\
& l_C(t_k^j) \geq d(t_k^j),
\end{aligned} \qquad (9)$$

where $\Delta SOH_{\mathcal{P}}(t_k)$ represents the SOH reduction of the battery pack at time slot t_k; I_{max}^+ and I_{min}^- represent the discharge current and charge current thresholds, respectively; SOC_{min} and SOC_{max} indicate the lower and upper bounds of the SOC, respectively, which are required to prevent excessive discharging and charging; $l_D(t_k^j)$ and $l_C(t_k^j)$ represent the power load in cycle j up until time slot t_k when discharging and charging, respectively; and $d(t_k^j)$ indicates the power demand at time slot t_k in cycle j. The SOH reduction of the battery pack at time t_k ($\Delta SOH_{\mathcal{P}}(t_k)$) is defined as

$$\Delta SOH_{\mathcal{P}}(t_k) = SOH_{\mathcal{P}}(t_{k-1}) - SOH_{\mathcal{P}}(t_k), \qquad (10)$$

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

$$\Delta SOH_{\mathcal{P}}(t_k) \geq 0. \qquad (11)$$

3. The Proposed Algorithm

To tackle the optimization problem (9), we propose a battery-scheduling algorithm that is run by the BMS. In each time slot, the algorithm first collects measurement data that include the terminal voltage, current, and temperature of each cell, then estimates the SOC and the SOH (Algorithm 1) and controls the charging or discharging process of the BESS based on the load demand (Algorithm 2). Algorithms 1 and 2 return a state vector consisting of a set of SOC values of cells ($\mathcal{C}(t_k)$) a set of SOH values of cells ($\mathcal{H}(t_k)$), as well as the battery pack current ($I_{\mathcal{P}}(t_k)$) and power demand ($d(t_k)$), triggering the DRL-based battery-scheduling algorithm (Algorithm 3). The overall flow of the proposed algorithm is shown in Figure 3. Each part of the proposed algorithm is discussed in detail in the subsections below.

Algorithm 1 EKF-based SOC and SOH estimation

1: Input: Measurement data $\mathcal{V}(t_k), \mathcal{I}(t_k), \mathcal{T}(t_k)$; Data tables
2: Output: $\mathcal{C}(t_k), \mathcal{H}(t_k)$
3: Estimate state vector $\hat{x}_i(t_k)$ and error covariance $\hat{P}_i(t_k)$ using (12) and (13)
4: Estimate terminal voltage $\hat{V}_i(t_k)$ and compute Kalman gain $G_i(t_k)$ using (17) and (20)
5: Update $x_i(t_k)$ and $P_i(t_k)$ using (21) and (22)
6: Update $SOC_i(t_k)$ and $M_i(t_k)$
7: **if** cycle is completed **then**
8:   Update $SOH_i(t_k)$ using (23)
9: **else**
10:   $SOH_i(t_k) \leftarrow SOH_i(t_{k-1})$
11: **end if**

Algorithm 2 The Charge/Discharge Control Algorithm

1: Input: $I_{\mathcal{P}}(t_k), l_D(t_k^j), l_C(t_k^j), d(t_k^j)$
2: Output: Discharge or Charge
3: **if** $I_{\mathcal{P}}(t_k) > 0$ and $t_k \in$ cycle j **then**   ▷ Discharging
4:   **if** $l_D(t_k^j) \geq d(t_k^j)$ **then**
5:     Convert discharge to charge
6:   **else**
7:     Continue to discharge
8:   **end if**
9: **else if** $I_{\mathcal{P}}(t_k) < 0$ and $t_k \in$ cycle j **then**   ▷ Charging
10:   **if** $l_C(t_k^j) \geq d(t_k^j)$ **then**
11:     Convert charge to discharge   ▷ cycle $j+1$
12:   **else**
13:     Continue to charge
14:   **end if**
15: **end if**

Algorithm 3 The Deep Q Network Switches Scheduling Algorithm

1: **Input:** state vector $s(t_k)$
2: **Output:** Optimal schedule action $\mathcal{X}(t_k)$
3: Initialize Replay experience $\mathcal{E}$ with capacity M
4: Add $\langle s(t_{k-1}), \mathcal{X}(t_{k-1}), r(t_{k-1}), s(t_k) \rangle$ into $\mathcal{E}$
5: Construct main network $\mathcal{Q}$ and target network $\bar{\mathcal{Q}}$
6: Initialize $\mathcal{Q}$ and $\bar{\mathcal{Q}}$ with random weights
7: Perform a gradient descent to minimize loss function $\mathcal{L}(\phi(t_k))$
8: **if** $s(t) \in \mathcal{K}$ **then**
9:   Select action $\mathcal{X}(t_k)$ using (30)   ▷ Switch ON/OFF
10: **else**
11:   Select action $\mathcal{X}(t_k)$ randomly
12: **end if**
13: Compute immediate reward $R(s(t_k), \mathcal{X}(t_k))$ using (31)
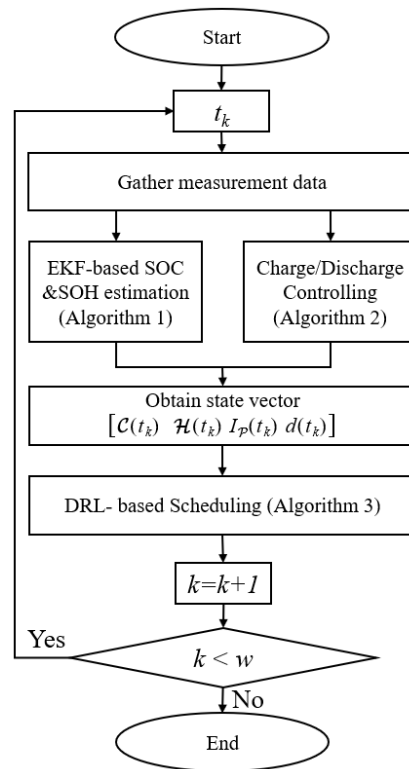14: Compute cumulative reward $r(t_k)$ using (32)

Figure 3. Overall flow chart of the proposed algorithm.

3.1. EKF-Based SOC and SOH Estimation

The algorithm estimates the SOC and SOH of each cell in the battery pack to observe the states of the battery cells using a third-order EKF. To obtain the SOC and SOH of battery cell i at t_k, the algorithm first estimates state vector $\hat{x}_i(t_k)$ and error covariance $\hat{P}_i(t_k)$ as

$$\hat{x}_i(t_k) = A_i(t_{k-1})x_i(t_{k-1}) + B_i(t_{k-1})I_i(t_{k-1}), \tag{12}$$

and

$$\hat{P}_i(t_k) = A_i(t_{k-1})P_i(t_{k-1})A_i(t_{k-1})^T, \tag{13}$$

where $x_i(t_{k-1})$ is the state vector of cell i at time $_{k-1}$, which is defined as

$$x_i(t_{k-1}) = [SOC_i(t_{k-1}), \; V_{pi}(t_{k-1}), \; 1/M_i(t_{k-1})]^T, \tag{14}$$

and $A_i(t_{k-1})$ and $B_i(t_{k-1})$ denote the transition matrix and the input matrix, respectively, which are defined as follows

$$A_i(t_{k-1}) = \begin{bmatrix} 1 & 0 & -\eta\Delta t I_i(t_{k-1}) \\ 0 & e^{-\frac{\Delta t}{R_{pi}(t_{k-1})C_{pi}(t_{k-1})}} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{15}$$

$$B_i(t_{k-1}) = \begin{bmatrix} 0 \\ R_{pi}(t_{k-1})\left(1 - e^{-\frac{\Delta t}{R_{pi}(t_{k-1})C_{pi}(t_{k-1})}}\right) \\ 0 \end{bmatrix}, \tag{16}$$

where $I_i(t_{k-1})$ is the measured current of cell i at t_{k-1}. $R_{si}(t_{k-1})$, $R_{pi}(t_{k-1})$, and $C_{pi}(t_{k-1})$ are functions of SOC_i, SOH_i, and T_i, respectively, which are obtained from two-dimensional look-up tables (A dataset [22] is used to construct look-up tables where $R_{si}(t_{k-1})$, $R_{pi}(t_{k-1})$, and $C_{pi}(t_{k-1})$ are exponential functions of $SOC_i(t_{k-1})$, such as $\left(x_1 \exp\left(x_2 SOC_i(t_{k-1})\right) + x_3\right)$,

and x_1, x_2, and x_3 are real numbers. These real numbers change when SOH_i decreases). Then, the algorithm estimates the terminal voltage ($\widehat{V}_i(t_k)$), using $\widehat{x}_i(t_k)$ and Jacobian matrices $C_i(t_k)$ and $D_i(t_k)$ as

$$\widehat{V}_i(t_k) = C_i(t_k)\widehat{x}_i(t_k) + D_i(t_k)I_i(t_k) \tag{17}$$

$$C_i(t_k) = \left[\frac{\delta V_{Oi}(t_k)}{\delta SOC_i(t_k)} \quad -1 \quad 0 \right] \tag{18}$$

$$D_i(t_k) = -R_{si}(t_k), \tag{19}$$

where the open-circuit voltage ($V_{Oi}(t_k)$) is identified by exploiting the look-up tables ($V_{Oi}(t_k)$ is the ath-order polynomial function of $SOC_i(t_k)$, which is defined as $\left(\sum_{b=0}^{a} y_b \left(SOC_i(t_k) \right)^b \right)$, where y_b is a real number that changes when SOH_i decreases). The algorithm calculates the Kalman gain ($G_i(t_k)$) to determine the error between the real, measured value and the estimated value using (13) as

$$G_i(t_k) = \widehat{P}_i(t_k)C_i(t_k)^T \left(C_i(t_k)\widehat{P}_i(t_k)C_i(t_k)^T \right)^{-1}. \tag{20}$$

Based on the estimated terminal voltage ($\widehat{V}_i(t_k)$), estimated state vector ($\widehat{x}_i(t_k)$), Kalman gain ($G_i(t_k)$), and measured terminal voltage ($V_i(t_k)$), the algorithm obtains the correct state vector ($x_i(t_k)$) as

$$x_i(t_k) = \widehat{x}_i(t_k) + G_i(t_k)\left(V_i(t_k) - \widehat{V}_i(t_k) \right). \tag{21}$$

Similarly, the algorithm corrects error covariance ($P_i(t_k)$) as

$$P_i(t_k) = \left(1 - G_i(t_k)C_i(t_k) \right)\widehat{P}_i(t_k). \tag{22}$$

From corrected state vector ($x_i(t_k)$), the proposed algorithm obtains $SOC_i(t_k)$ and $M_i(t_k)$. The algorithm updates the SOH of cell i after one cycle (complete charging and discharging of the battery pack), since the SOH does not decrease after one or several time slots [23]. The algorithm updates the SOH of cell i at time slot t_k as

$$SOH_i(t_k) = \begin{cases} \frac{\overline{M}_i^j(t_k)}{M_{new}} & \text{if cycle } j \text{ is completed;} \\ SOH_i(t_{k-1}) & \text{otherwise} \end{cases} \tag{23}$$

where $\overline{M}_i^j(t_k)$ is the effective current capacity (on average) of cell i in cycle j, which has $(k - \varkappa + 1)$ time slots if cycle j is completed at time slot t_k. The effective current capacity (on average) of cell i is calculated as

$$\overline{M}_i^j(t_k) = \frac{\sum_{\tau=\varkappa}^{k} M_i(t_\tau)}{k - \varkappa + 1}, \tag{24}$$

where cycle j starts at $t_\varkappa$ and ends at t_k. Algorithm 1 summarizes the EKF-based estimation for the SOH and SOC of cells.

3.2. The Charge/Discharge Control Algorithm

To control the process of charging and discharging the battery pack in the BESS, the algorithm first identifies the process that is underway. If the current of the battery pack is positive, i.e., $I_{\mathcal{P}}(t_k) > 0$, we calculate the amount of electric power discharged in cycle j $l_D(t_k^j)$ using (7). If $l_D(t_k^j)$ reaches electrical demand ($d(t_k^j)$), the BMS converts the BESS process from discharging to charging. Otherwise, the discharge process continues.

If $I_{\mathcal{P}}(t_k)$ is negative, the algorithm determines $l_C(t_k^j)$ (the amount of electrical power charged in cycle j) using (8) and compares it with electrical demand ($d(t_k^j)$). If $l_C(t_k^j)$ reaches $d(t_k^j)$, the algorithm converts the BESS process from charging to discharging for a new cycle ($j + 1$); otherwise, it continues charging. The process of charging and discharging the battery pack is summarized in Algorithm 2.

3.3. Deep Reinforcement Learning-Based Scheduling Algorithm

A deep Q network (DQN) scheduling algorithm is proposed for the ON/OFF cell switches in the battery pack. The scheduling algorithm has three elements: state $s(t_k)$, which represents the current state of the BESS; action $\mathcal{X}(t_k)$, which indicates cell switches that are ON or OFF; and reward function $r(t_k)$ based on action $\mathcal{X}(t_k)$. The algorithm selects action $\mathcal{X}(t_k)$ by interacting with the environment, i.e., the BESS, to perceive the state of the battery pack ($s(t_k)$) to maximize the cumulative reward ($r(t_k)$), i.e., to minimize SOH reduction of the battery pack. To choose an optimal schedule as $\mathcal{X}(t_k)$ for state $s(t_k)$, the algorithm utilizes and updates acquired knowledge ($\mathcal{K}$) using deep reinforcement learning. That knowledge includes a switch-scheduling policy for the given battery states and the corresponding scheduling of rewards. The DQN-based scheduling algorithm is summarized in Algorithm 3.

The algorithm first observes the current environmental state of the battery pack and obtains state vector $s(t_k)$, which is defined as

$$s(t_k) = \big[\mathcal{C}(t_k), \mathcal{H}(t_k), I_{\mathcal{P}}(t_k), d(t_k) \big], \tag{25}$$

where $\mathcal{C}(t_k)$ and $\mathcal{H}(t_k)$ are sets of the SOCs and SOHs of N cells, respectively; $I_{\mathcal{P}}(t_k)$ is the load current of the battery pack; and $d(t_k)$ is the load demand. Then, the algorithm initializes knowledge ($\mathcal{K}$) that includes replay experience ($\mathcal{E}$) with samples $\langle s(t_{k-1}), \mathcal{X}(t_{k-1}), r(t_{k-1}), s(t_k) \rangle$, a main network ($\mathcal{Q}$), and a target network ($\bar{\mathcal{Q}}$) with random weights. Neural networks $\mathcal{Q}$ and $\bar{\mathcal{Q}}$ have the same structure. The algorithm explores actions based on past experiences to update the acquired knowledge that leads to a long-term benefit. The DQN updates acquired knowledge ($\mathcal{K}$) by minimizing loss function $\mathcal{L}(\phi(t_k))$ using gradient descent. The loss function is defined as

$$\mathcal{L}\big(\phi(t_k)\big) \leftarrow \mathbb{E}\bigg[\big(\bar{Q}(t_{k-1}) - Q(t_{k-1}) \big)^2 \bigg], \tag{26}$$

which $\phi(t_k)$ is the DQN network parameter (weight of the main network) and is calculated as

$$\phi(t_k) = \phi(t_{k-1}) + \alpha \nabla \mathcal{L}\big(\phi(t_{k-1})\big) \tag{27}$$

where $\alpha \in (0, 1]$ is the learning factor. $Q(t_{k-1})$ shows the expected discounted cumulative reward after time slot t_{k-1} in main network $\mathcal{Q}$, and $\bar{Q}(t_{k-1})$ is the target action value of the target network ($\bar{\mathcal{Q}}$), which represents the maximum cumulative reward, i.e., the minimum SOH reduction for the battery pack. $Q(t_{k-1})$ and $\bar{Q}(t_{k-1})$ are calculated as

$$
\begin{aligned}
Q(t_{k-1}) &= Q\big(s(t_{k-1}), \mathcal{X}(t_{k-1}) \mid \phi\big) \\
&= \mathbb{E}\big[r(t_{k-1}) \mid s(t_{k-1}), \mathcal{X}(t_{k-1}) \big],
\end{aligned}
\tag{28}
$$

and

$$\bar{Q}(t_{k-1}) = r(t_{k-1}) + \gamma \max_{\mathcal{X}(t_k)} Q\big(s(t_k), \mathcal{X}(t_k) \mid \bar{\phi}\big), \tag{29}$$

where $\gamma \in (0, 1]$ is the discount cumulative factor indicating the degree of emphasis of future rewards, and $\phi = \{\phi(t_1), \phi(t_2), ..., \phi(t_k)\}$ and $\bar{\phi} = \{\phi(\bar{t}_1), \phi(\bar{t}_2), ..., \phi(\bar{t}_k)\}$ represent the weights of networks $\mathcal{Q}$ and $\bar{\mathcal{Q}}$, respectively. After determining the loss based on an action, the target network ($\bar{\mathcal{Q}}$) copies the weight of the main network ($\mathcal{Q}$), i.e., $\bar{\phi} = \phi$.

To utilize the past experience in a DQN-based scheduling algorithm, the proposed algorithm looks at the acquired knowledge ($\mathcal{K}$) to determine whether state $s(t_k)$ is in $\mathcal{K}$ or not. If state $s(t_k)$ is in $\mathcal{K}$, the algorithm chooses action $\mathcal{X}(t_k)$ based on an ϵ-greedy policy, i.e., it chooses a random action with probability $p = \epsilon$ or the action with probability $p = 1 - \epsilon$ that has the largest value for $Q(s(t_k), \mathcal{X}(t_k))$. Based on the ϵ-greedy policy [24], action $\mathcal{X}(t_k)$ is defined as

$$\mathcal{X}(t_k) = \begin{cases} \text{random action, with } p = \epsilon \\ \arg\max_{\mathcal{X}(t_k)} Q(s(t_k), \mathcal{X}(t_k) \mid \phi), \text{with } p = 1 - \epsilon \end{cases} \tag{30}$$

In the case in which state $s(t_k)$ is not in $\mathcal{K}$, scheduling action $\mathcal{X}(t_k)$ is performed at random. After taking action $\mathcal{X}(t_k)$ based on observed state $s(t_k)$, the algorithm evaluates the immediate reward as

$$R(s(t_k), \mathcal{X}(t_k)) = E[-\Delta SOH_{\mathcal{P}}(t_k)]. \tag{31}$$

Then, the algorithm determines the cumulative reward ($r(t_k)$) by interacting with the environment and looks for an optimal policy to maximize $r(t_k)$. The cumulative reward ($r(t_k)$) is calculated as

$$r(t_k) = \mathbb{E}\left[\sum_{h=k}^{w} \gamma^h R(s(t_h), \mathcal{X}(t_h))\right]. \tag{32}$$

The algorithm minimizes loss function $\mathcal{L}(\phi(t_k))$ so that action value $Q(t_{k-1})$ has the same value as target action value $\bar{Q}(t_{k-1})$, which also means that the SOH of the battery pack is optimized. The DQN-based scheduling algorithm is summarized in Algorithm 3, and the DQN training process is shown in Figure 4.
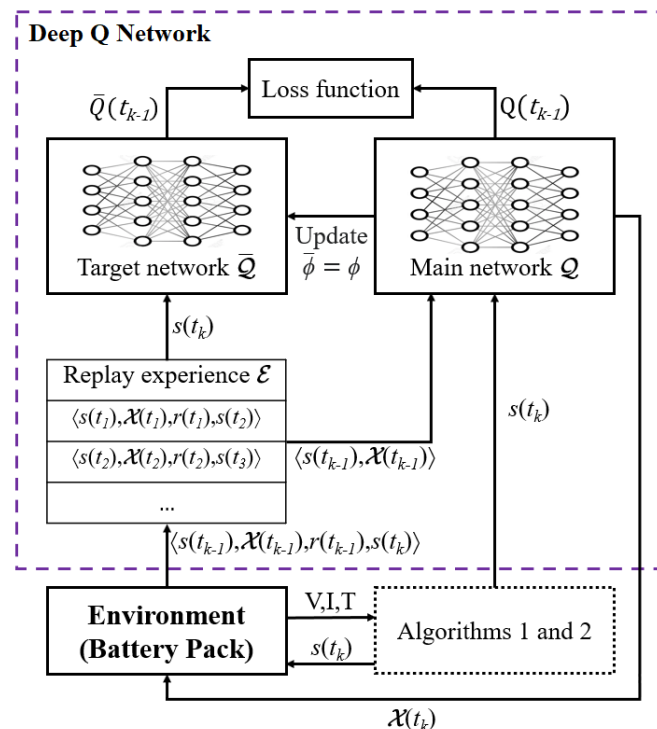

Figure 4. The training process in the DQN.

4. Performance Evaluation

4.1. Simulation Environment

The simulation was conducted using a lithium-ion battery model and was implemented in MATLAB and Simulink R2022a. To evaluate the performance of the proposed algorithm, we consider a parallel-connected battery pack including four lithium 3.7 V/2.2 Ah batteries with heterogeneous states of health (90.01%, 86.77%, 84.13%, and 78.15% corresponding to cells 1 to 4, respectively). MOSFETs with low ON resistance and low power are installed to connect and disconnect the battery cells from the battery pack. We consider different power demand conditions to evaluate the effectiveness of the algorithm. Based on the maximum capacity of a battery pack with new battery cells, we obtain a dynamic power demand profile by generating values from a uniform distribution across 20% to 60% of the maximum energy of a battery pack (i.e., between 6.51 Wh and 19.54 Wh). For the constant power demand, we calculate the mean value of the dynamic power demand profile as

$$D_{avg} = \frac{1}{W} \sum_{k=1}^{w} d(t_k), \tag{33}$$

where $d(t_k)$ is the power demand at time slot t_k, and W is the number of time slots during working time ($\mathcal{W} = \{t_k \mid k = 1, 2, ..., w\}$). Figure 5 shows dynamic and constant power demand profiles. Constant power demand is equal to 13.13 Wh (i.e., 40.32% of the maximum energy of a new battery pack). We set the load current of the battery pack when discharging and charging to 8 A.

A dataset compiled by NASA [22] was used to model a first-order Thévenin equivalent battery model with a reduction in SOH. We also use the dataset to obtain actual SOC and SOH values, which are compared with the estimated values. The dataset includes 28 lithium cobalt oxide 18,650 cells with a nominal capacity of 2.2 Ah, including in-cycle measurements of terminal voltage, current, and cell temperature. The dataset also includes measurements for discharging capacity and EIS impedance readings. We identify the EIS parameters, which include V_{Oi}, R_{si}, R_{pi}, and C_{pi}, in the 90% to 60% SOH range using the dataset.
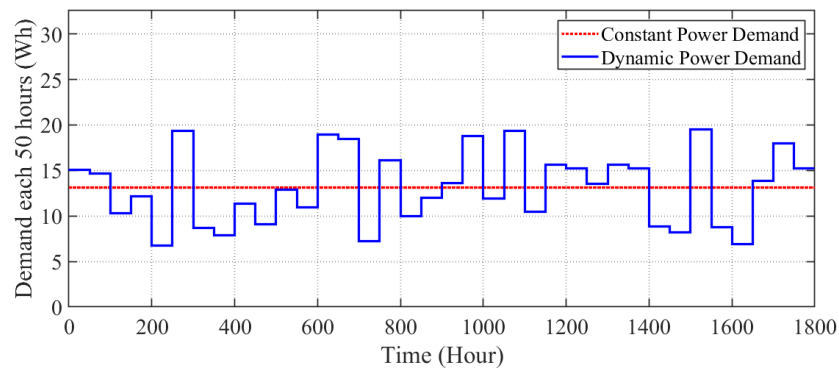

Figure 5. Load demand profile.

The structure of neural networks includes one 10-dimension input layer, two 256-dimension hidden layers, one 256-dimension LSTM layer, and one 16-dimension output layer. The input layer consists of 10 elements of the battery state ($s(t)$), since there are four battery cells in a battery pack. The output layer consists of 11 cases (There must be at least two batteries ON at the same time, since we consider $8A$ current during discharging and the maximum output current of one battery is 4 A) of schedule action $\mathcal{X}(t)$. We set the learning rate (α) to 0.001, the ϵ-greedy value to 0.9, and the discount factor (γ) to 0.99. The period of the target network update is 10 time steps. Other simulation parameters are summarized in Table 2.

Table 2. Simulation parameters.

Parameter	Value
Number of battery cells	4
Battery type	Lithium 3.7 V/2.2 Ah
Total capacity (new)	32.56 Wh
Constant power demand	13.13 Wh (40.32%)
$I_{discharge}$	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 ($\mathcal{W}$)	1800 h
Δt	10 min
Capacity M of experience $\mathcal{E}$	500 slots
Learning rate (α)	0.001
ϵ-greedy	0.9
Discount factor (γ)	0.99
Period of target network update	10 time slots

For the performance evaluation, we first verify the accuracy of the estimation algorithm by determining the error between estimated and actual values. Then, we investigate the effect of the proposed algorithm on the lifetime of a battery pack and the SOHs of the cells under dynamic and constant loads. To validate the performance of the proposed algorithm, we compare it with methods proposed in previous works, including an enhanced Coulomb counting method [7], a hybrid statistical data-driven estimation method [11], and a multi-actor–critic scheduling algorithm [6]. For comparison, we combine the scheduling and estimation algorithms and obtain the BESS performance. We also compare the proposed estimation algorithm with the enhanced Coulomb counting method and the hybrid statistical data-driven estimation method. For the sake of simplicity, we denote the proposed third-order extended Kalman filter (EKF) estimation algorithm as EKFest, the proposed deep Q network scheduling algorithm as DQNsch, the multi-actor–critic scheduling algorithm as MACsch, the hybrid statistical data-driven estimation method as DDest, the enhanced Coulomb counting method as ECest, and simulations without any scheduling algorithm as Non Schedule.

4.2. State Estimation Verification

To evaluate the performance results of the proposed algorithm in estimating the SOC and SOH for each cell, we first show the estimated terminal voltage of each cell in a battery pack. Figure 6 shows the root mean square error (RMSE) between the measured terminal voltage and the estimated terminal voltage. The RMSE between the measured and estimated values of the terminal voltage for each cell is close to 0.01 V and remains small over time. The small difference between measured and estimated terminal voltages shows that the proposed algorithm accurately models terminal voltage, which leads to a more accurate estimation of the SOC and SOH of a cell.

The performance results of the proposed algorithm in estimating the SOC and SOH for each cell in terms of RMSE and mean absolute error (MAE), respectively, are shown in Figure 7. The proposed estimation algorithm has the lowest RMSE compared to other works in estimating the SOCs of cells, as shown in Figure 7a. The RMSE between the actual and estimated values of the SOC for each cell is close to 1% under the proposed algorithm. The error of the proposed algorithm in estimating the SOHs of the cells is shown in Figure 7b. The proposed algorithm has an error of less than 0.2% for SOH, which is 50% less than the other estimation algorithms. ECest shows the worst performance, degrading over time. Note that the performance of the proposed estimation algorithm becomes more stable over time. Estimating the SOC and SOH of the cells with low error is

Energy Conversion Circuit Laboratory

of great significance in order to obtain optimal ON/OFF cell scheduling that extends the lifetime of a battery pack.
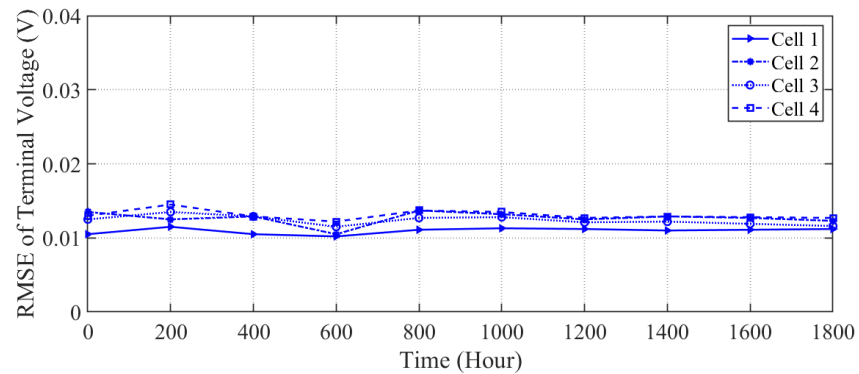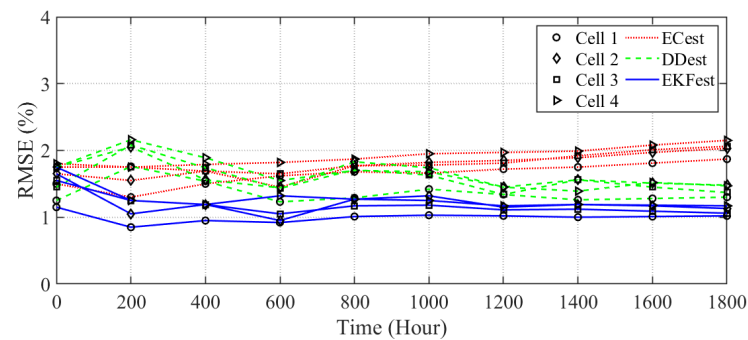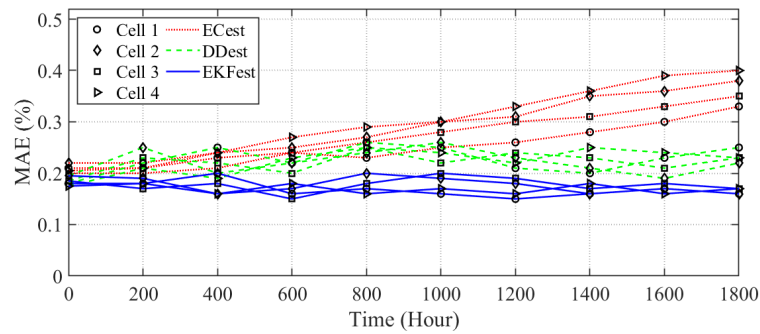

Figure 6. Root mean square error between actual and estimated terminal voltage using the proposed algorithm.

(**a**)

(**b**)

Figure 7. State estimation evaluation: (**a**) root mean square error of SOC estimation and (**b**) mean absolute error of SOH estimation.

4.3. Impact of the Proposed Algorithms on Battery Pack Lifetime

The impact of the proposed algorithm on battery pack lifetime in terms of SOH reduction under constant and dynamic power demands is evaluated and shown in Figure 8. The proposed algorithm achieves better performance under both constant and dynamic power demands compared to other algorithms. The proposed algorithm reduces the SOH decay in the battery pack by efficiently scheduling the ON/OFF switching of the cells based on accurate estimation of SOHs and SOCs, resulting in an increase in battery pack lifetime.
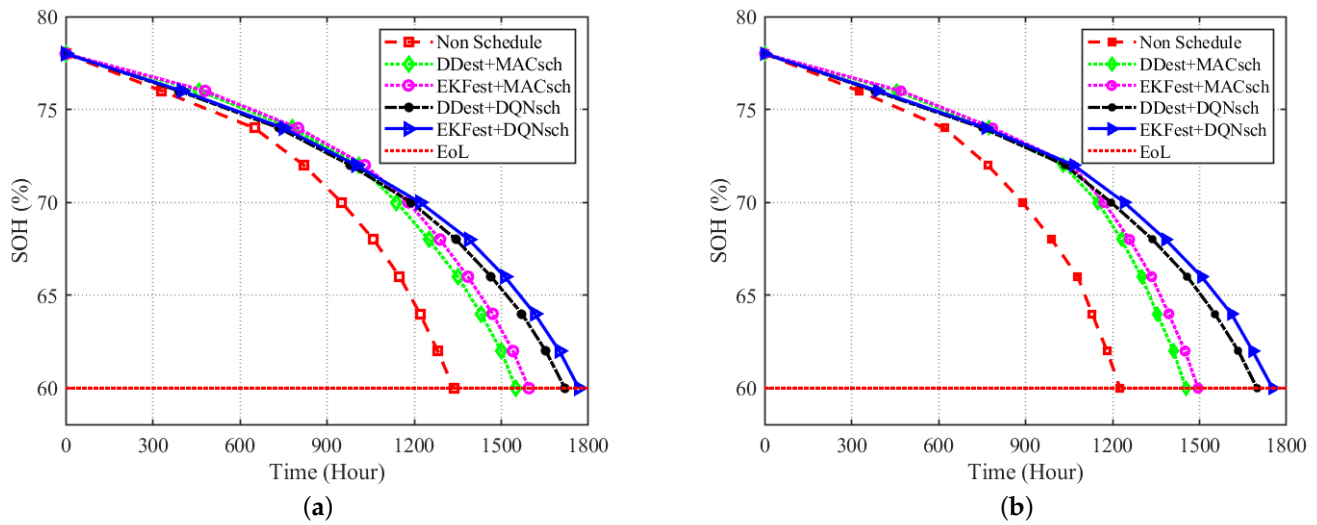
Figure 8. SOH reduction of the battery pack under (**a**) constant power demand and (**b**) dynamic power demand.

The SOH of the battery pack reaches 60% (the end of its second life (EoL)) after a working time of 1767 h under constant power demand, which represents a 13.9% increase in battery pack lifetime compared to previous work (DDest + MACsch). Under dynamic power demand, battery pack lifetime also increases by 20.6% under the proposed algorithm compared to previous work. In addition, the difference in the performance of the proposed algorithm under constant and dynamic power demand is quite small, but the performance of methods proposed in previous work degrades under dynamic power demand. Hence, the proposed algorithm can hence efficiently schedule ON/OFF switching of battery cells to adapt to dynamic power demand.

Compared to DDest + MACsh, the lifetime of the battery pack is higher under EKFest + MACsch and DDest + DQNsch. This shows that the proposed estimation algorithm, as well as the scheduling algorithm, can an impact in extending the lifetime of a battery pack. DDest + DQNsch achieves better performance than EKFest + MACsch, which means optimal scheduling is a more dominant factor in prolonging battery pack lifetime. MACsch achieves worse performance, since it does not consider SOC while scheduling the ON/OFF cell switches to meet power demand. Without scheduling (Non-Schedule), the lifetime of the battery pack reduces rapidly because the weakest cell, i.e., the cell with the lowest SOH, operates continuously.

4.4. Impact of the Proposed Algorithm on Capacity Balancing

The effectiveness of the proposed algorithm in balancing the SOH of cells under constant and dynamic load demands is shown in Figures 9 and 10, respectively. Without a scheduling algorithm (Non-Schedule), all the cells in the battery pack are utilized all the time, irrespective of their SOC and SOH, resulting in imbalanced states of health and increasing SOH reduction in the battery pack, irrespective of load demand conditions, as shown in Figures 9a and 10a.

All the algorithms balance the SOH of cells in the battery pack under constant and dynamic load demands, as shown in Figure 9b–e and Figure 10b–e, respectively. Even though the methods proposed in other works achieve SOH balancing among battery pack cells, battery lifetime (the SOH of each cell) decreases rapidly under the other algorithms compared to the proposed algorithm. This means that with heterogeneous states of health for cells in a battery pack, the proposed algorithm offers better performance than other algorithms by extending the second life of battery cells. All the algorithms achieve SOH

standard deviations close to zero by balancing the capacity of each cell over time under constant power demand, which can be seen in Figure 9f.

Under dynamic load demand, EKFest + DQNsch achieves more even SOH balancing and reduces the standard deviation of the cells' SOHs to zero, while other algorithms fail to balance the SOHs of cells, except for the DDest + DQNsch, which achieves the second-best performance, as shown in Figure 10b–f. The SOH of the weakest cell (cell 4, which has the lowest initial SOH) reaches 60%, while other cells have SOHs of more than 60% under algorithms proposed in other works, resulting in higher standard deviations and earlier end of second life of the battery pack. DDest + DQNsch reduces the standard deviation of SOHs and extends battery life compared to other scheduling algorithms. This shows the effectiveness of the proposed scheduling algorithm in managing a parallel-connected BESS, even with a less accurate estimation algorithm. The superior performance of the proposed algorithm under the different load demand conditions shows the robustness of the algorithm to load demands.
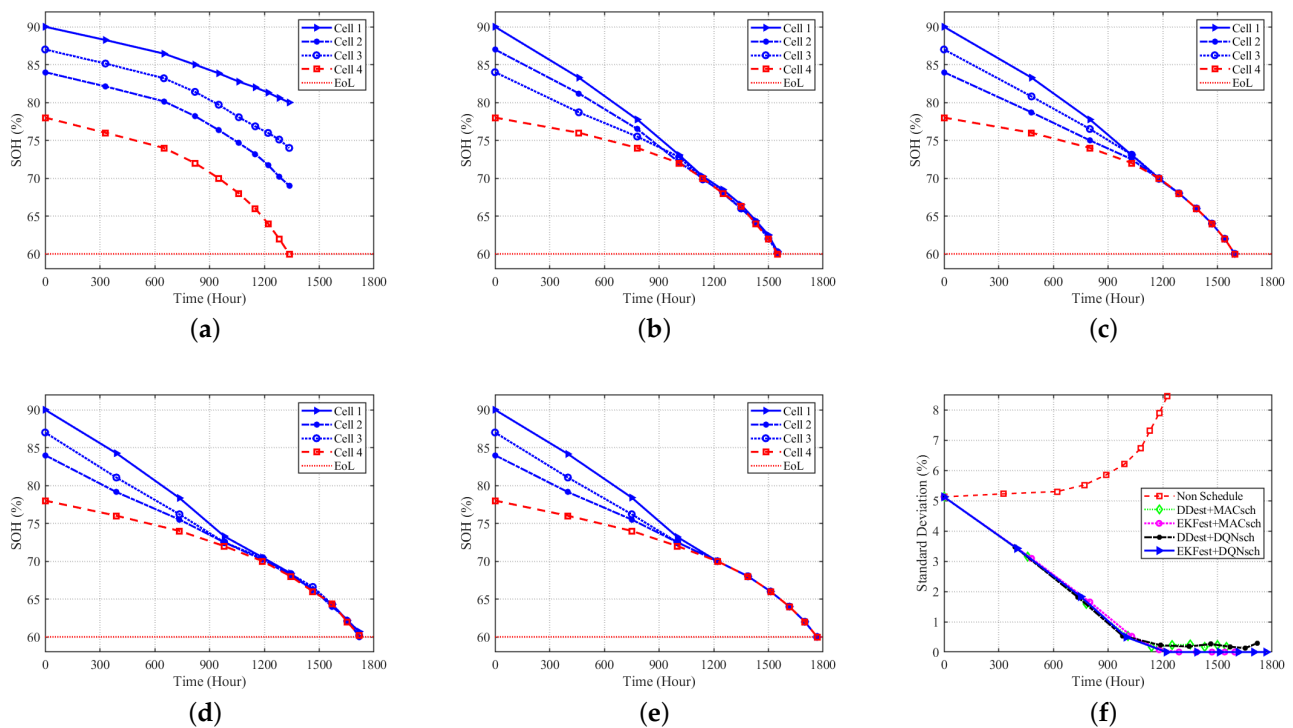
Figure 9. SOH balancing under constant power demand with (**a**) Non-Schedule, (**b**) DDest + MACsch, (**c**) EKFest + MACsch, (**d**) DDest + DQNsch, (**e**) EKFest + DQNsch, and (**f**) the standard deviation of SOHs among the cells.

4.5. Impact of Numbers of Batteries on the Proposed Algorithm

We study the impact of the number of parallel-connected batteries for the BESS on the proposed algorithm under dynamic load demand according to the SOH profiles shown in Table 3. The SOH profiles of batteries have the same SOH average (84.77%) and standard deviation (5.02%).

Table 3. SOHs of batteries.

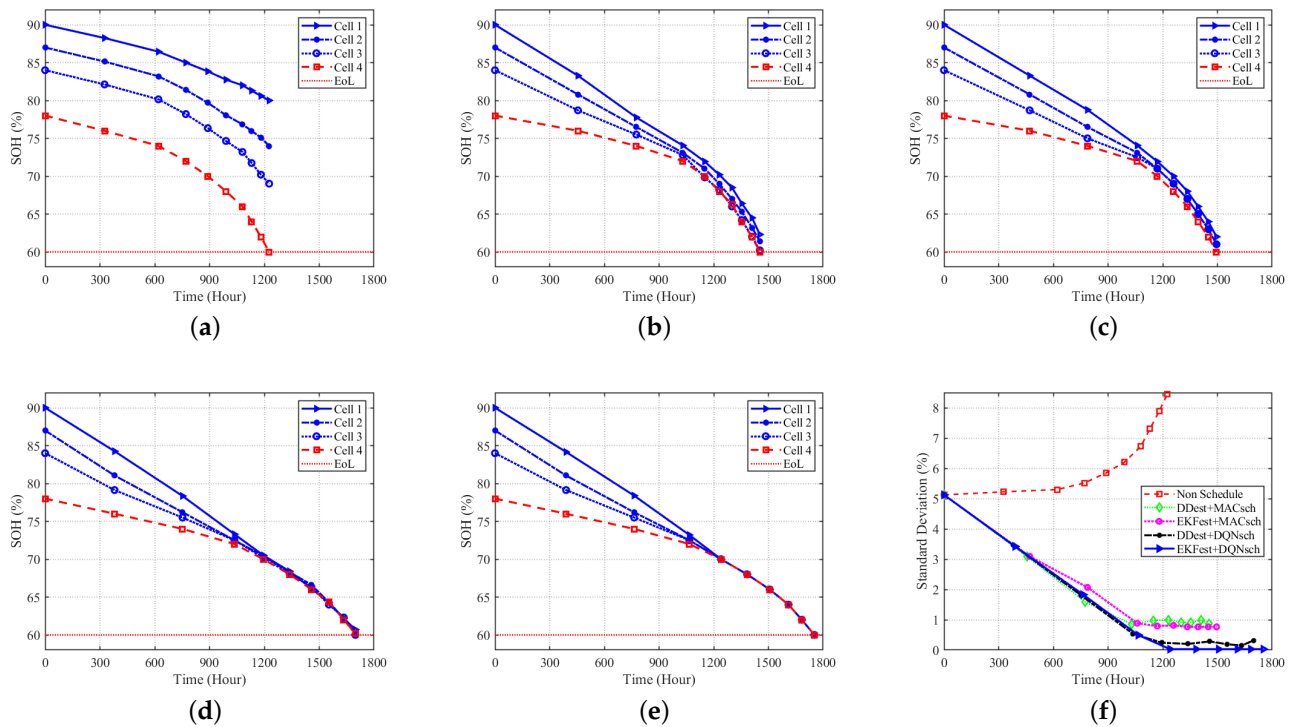| Number of Batteries	SOH Profile (%)	Total Max. Capacity
3	89.59, 85.14, 79.57	5.59 Ah
4	90.01, 86.77, 84.13, 78.15	7.46 Ah
5	91.05, 87.95, 84.76, 81.95, 78.15	9.32 Ah
6	91.17, 90.05, 84.86, 82.67, 81.65, 78.21	11.19 Ah

Figure 10. SOH balancing under dynamic power demand with (**a**) Non-Schedule, (**b**) DDest + MACsch, (**c**) EKFest + MACsch, (**d**) DDest + DQNsch, (**e**) EKFest + DQNsch, and (**f**) the standard deviation of SOHs among the cells.

The performance of the proposed algorithm under different battery conditions in terms of the operational working time and standard deviation in SOHs is shown in Figure 11. The proposed algorithm (EKFest + DQNsch) achieves higher operational time (i.e., extends the second life of a battery pack) compared to other algorithms, as can be seen in Figure 11a. The proposed algorithm minimizes the SOH reduction of the battery pack in each time slot by balancing the SOHs of battery cells, thereby extending the battery pack's lifetime.
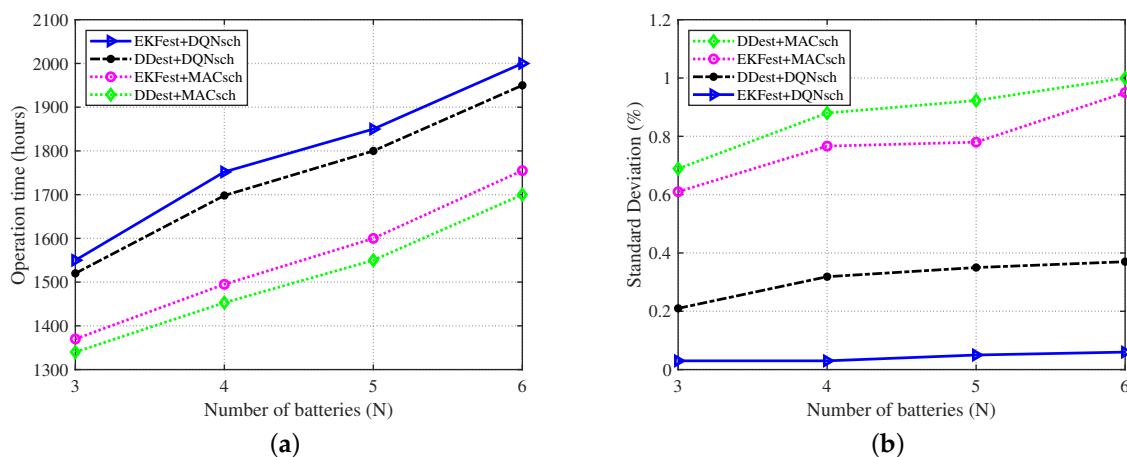
Figure 11. Performance of the scheduling algorithms with different numbers of batteries under dynamic power demand: (**a**) operation time of the battery pack until the SOH reaches 60% and (**b**) standard deviation of SOHs among the batteries.

The proposed algorithm achieves the lowest standard deviation with different numbers of batteries in a battery pack, as shown in Figure 11b. The standard deviation in

SOHs increases by a minimal amount under the proposed algorithm with an increase in the number of batteries compared to other algorithms. The combinations of the proposed estimation and the proposed scheduling algorithms with the algorithms proposed in previous works (EKFest + MACsch and DDest + DQNsch) increase the lifetime of a battery pack and achieve a more uniform SOH balance compared to the combination of previously proposed algorithms (i.e., DDest + MACsch). This shows the effectiveness of both parts of the proposed algorithm in the optimization of BESSs. Figure 11 shows that the proposed algorithm is robust to the number of battery cells in a battery pack in a BESS.

5. Conclusions and Future Work

In this paper, we proposed a DRL-based battery management algorithm to optimize battery lifetime for retired batteries with heterogeneous SOHs in a parallel-connected BESS. The proposed algorithm

(i) estimated the SOCs and SOHs of all battery cells using EKF;
(ii) used estimated SOCs and SOHs to represent the state of a BESS for DRL-based scheduling; and
(iii) controlled the ON/OFF switches of battery cells inside the battery pack utilizing deep Q network knowledge.

Via simulation, we showed that the proposed algorithm outperformed other proposed algorithms by showing lower estimation errors for battery cell states and extending the battery pack's second life. The proposed algorithm extended the operation time of the battery pack by 13.9% and 20.6% compared to other algorithms under constant and dynamic power demand, respectively.

Regarding future work, we will consider a BESS in which multiple battery packs are connected in series and each battery pack has parallel-connected battery cells. Such a configuration leads to high dimensions of state space. Furthermore, the deployment of smart-grid technologies that include energy storage systems [25] requires hundreds of battery cells connected in parallel or in series in a BESS. In such systems, DRL-based battery management algorithms can achieve limited performance due to high-dimensional state space. We will investigate a distributed reinforcement learning approach to counter the limitations of centralized approaches for large-scale energy storage systems. Additionally, an experimental setup will be considered to observe the impact of the battery management algorithm on real systems.

References

1. Marinaro, M.; Bresser, D.; Beyer, E.; Faguy, P.; Hosoi, K.; Li, H.; Sakovica, J.; Amine, K.; Wohlfahrt-Mehrens, M.; Passerini, S. Bringing forward the development of battery cells for automotive applications: Perspective of R&D activities in China, Japan, the EU and the USA. *J. Power Sources* **2020**, *459*, 228073.
2. Ding, Y.L.; Cano, Z.; Yu, A.; Lu, J.; Chen, Z. Automotive Li-Ion Batteries: Current Status and Future Perspectives. *Electrochem. Energy Rev.* **2019**, *2*, 1–28. [CrossRef]
3. Hunt, G. *USABC Electric Vehicle Battery Test Procedures Manual. Revision 2*; USDOE: Washington, DC, USA, 1996.
4. I.E.A. *Global EV Outlook 2019*; International Energy Agency: Paris, France, 2019.
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. Sui, Y.; Song, S. A Multi-Agent Reinforcement Learning Framework for Lithium-ion Battery Scheduling Problems. *Energies* **2020**, *13*, 1982. [CrossRef]
7. Ng, K.S.; Moo, C.S.; Chen, Y.P.; Hsieh, Y.C. Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. *Appl. Energy* **2009**, *86*, 1506–1511. [CrossRef]
8. Zhao, L.; Lin, M.; Chen, Y. Least-squares based coulomb counting method and its application for state-of-charge (SOC) estimation in electric vehicles. *Int. J. Energy Res.* **2016**, *40*, 1389–1399. [CrossRef]
9. Yang, Y.; Zhao, L.; Yu, Q.; Liu, S.; Zhou, G.; Shen, W. State of charge estimation for lithium-ion batteries based on cross-domain transfer learning with feedback mechanism. *J. Energy Storage* **2023**, *70*, 108037. [CrossRef]
10. Xiong, X.; Wang, Y.; Li, K.; Chen, Z. State of health estimation for lithium-ion batteries using Gaussian process regression-based data reconstruction method during random charging process. *J. Energy Storage* **2023**, *72*, 108390. [CrossRef]
11. Song, Y.; Liu, D.; Liao, H.; Peng, Y. A hybrid statistical data-driven method for on-line joint state estimation of lithium-ion batteries. *Appl. Energy* **2020**, *261*, 114408. [CrossRef]
12. Plett, G.L. Extended Kalman filtering for battery management systems of LiPB-based HEV battery packs: Part 3. State and parameter estimation. *J. Power Sources* **2004**, *134*, 277–292. [CrossRef]
13. Xiong, R.; Pan, Y.; Shen, W.; Li, H.; Sun, F. Lithium-ion battery aging mechanisms and diagnosis method for automotive applications: Recent advances and perspectives. *Renew. Sustain. Energy Rev.* **2020**, *131*, 110048. [CrossRef]
14. Zou, Y.; Hu, X.; Ma, H.; Li, S.E. Combined State of Charge and State of Health estimation over lithium-ion battery cell cycle lifespan for electric vehicles. *J. Power Sources* **2015**, *273*, 793–803. [CrossRef]
15. Song, C.; Shao, Y.; Song, S.; Chang, C.; Zhou, F.; Peng, S.; Xiao, F. Energy Management of Parallel-Connected Cells in Electric Vehicles Based on Fuzzy Logic Control. *Energies* **2017**, *10*, 404. [CrossRef]
16. Zhang, H.; Pei, L.; Sun, J.; Song, K.; Lu, R.; Zhao, Y.; Zhu, C.; Wang, T. Online Diagnosis for the Capacity Fade Fault of a Parallel-Connected Lithium Ion Battery Group. *Energies* **2016**, *9*, 387. [CrossRef]
17. Kim, H.; Shin, K.G. Scheduling of Battery Charge, Discharge, and Rest. In Proceedings of the 2009 30th IEEE Real-Time Systems Symposium, Washington, DC, USA, 1–4 December 2009; pp. 13–22.
18. Sun, B.; Xiong, L.; Liu, X.; Zhu, H. Research on Electromagnetic Compatibility in the Design of Battery Management System. In Proceedings of the 2023 IEEE International Conference on Mechatronics and Automation (ICMA), Harbin, China, 6–9 August 2023; pp. 363–368.
19. Bruen, T.; Marco, J.; Gama, M. Current Variation in Parallelized Energy Storage Systems. In Proceedings of the 2014 IEEE Vehicle Power and Propulsion Conference (VPPC), Coimbra, Portugal, 27–30 October 2014; pp. 1–6.
20. Kim, J.; Cho, B. Screening process-based modeling of the multi-cell battery string in series and parallel connections for high accuracy state-of-charge estimation. *Energy* **2013**, *57*, 581–599. [CrossRef]
21. 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]
22. 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.
23. 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]
24. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; A Bradford Book: Cambridge, MA, USA, 2018.
25. Giannelos, S.; Borozan, S.; Aunedi, M.; Zhang, X.; Ameli, H.; Pudjianto, D.; Konstantelos, I.; Strbac, G. Modelling Smart Grid Technologies in Optimisation Problems for Electricity Grids. *Energies* **2023**, *16*, 5088. [CrossRef]