Target-Mean State-of-Charge Control for Maximum Utilization of Heterogeneous Reconfigurable Battery Systems Under Constant-Bus Constraints

SZTUKA, Mateusz, MUSAMEH, Mohammad, ALI, Asma, RICHARDSON, Nicholas <http://orcid.org/0009-0005-0193-0079>, DI NUOVO, Alessandro <http://orcid.org/0000-0003-2677-2650> and ISSA, Walid <http://orcid.org/0000-0001-9450-5197>

batteries

MDPI

Article

Target-Mean State-of-Charge Control for Maximum Utilization of Heterogeneous Reconfigurable Battery Systems Under Constant-Bus Constraints

Mateusz Sztuka *, Mohammad Musameh, Asma Ali, Nicholas Richardson, Alessandro Di Nuovo and Walid Issa *

School of Engineering, Sheffield Hallam University, Sheffield S1 1WB, UK
* Correspondence: mattsztuka@gmail.com (M.S.); walid.issa@shu.ac.uk (W.I.)

Abstract

Cell degradation in second-life battery packs introduces heterogeneous capacity and internal resistance mismatch, reducing the effectiveness of conventional balancing approaches and limiting available pack runtime. Although equal state of charge (SoC) does not necessarily imply equal usable capacity, SoC-based control remains attractive for runtime-oriented operation. This paper proposes a target-mean controller for heterogeneous reconfigurable battery packs under constant-bus constraints that aims to improve runtime and achieve the cutoff-defined theoretical maximum capacity utilization limit. Using only real-time cell SoC measurements and legal switching actions, the controller selects the configuration that best reduces deviation from the pack-average SoC while preferentially loading cells above the mean. The online action selection requires no active balancing hardware, no explicit capacity or state of health (SoH) estimation, and no offline optimization; experimentally measured capacities are used only for calibrated Coulomb-counting SoC estimation. Simulation results on a heterogeneous five-cell reconfigurable battery pack show that the proposed controller reaches the cutoff-defined 90% theoretical utilization limit in the full-initial-SoC cases, while also extending runtime and reducing switching activity by up to 11.75% relative to the comparison methods. Hardware validation on a five-cell prototype further confirms this trend, achieving 89.12% experimental utilization, zero final SoC spread, and higher delivered energy than both comparison methods. A stepped-load hardware test further achieved 88.19% utilization from current integration, corresponding to 97.99% of the cutoff-defined 90% theoretical limit. The results suggest that, for heterogeneous second-life packs, SoC-based reconfiguration control can achieve both runtime improvement and near-maximum utilization without the added complexity of explicit SoH-aware balancing.

Keywords: reconfigurable battery systems; second-life batteries; state of charge; capacity utilization; runtime improvement; battery reconfiguration; battery management

Academic Editor: Jinhao Meng

Received: 20 May 2026
Revised: 15 June 2026
Accepted: 16 June 2026
Published: 18 June 2026

1. Introduction

Second-life lithium-ion batteries are attracting increasing interest for low-cost and sustainable energy storage applications [1,2]. Second-life battery deployment has been widely discussed as a route to extend battery lifetime, reduce waste, and improve the value extracted from EV battery production, although technical, economic, policy, and repurposing barriers remain [1]. However, the practical deployment of second-life

lithium-ion batteries depends strongly on reliable characterization at the repurposing stage. Braco et al. [2] showed that capacity and internal resistance are key indicators for assessing reused cells, modules, and packs, but conventional characterization can require long testing times and costly equipment.

Non-invasive characterization of commercial lithium-ion cells commonly involves capacity, impedance, resistance, and voltage-based measurements, but these procedures can become time-intensive when applied at scale [3].

Additionally, after first-life use, cell degradation introduces significant variation in capacity and internal resistance, which complicates pack-level control and reduces the available runtime and usable energy of the pack [2]. In series-connected strings, the weakest cell can dominate the usable capacity of the string during discharge, leading to early cutoff and preventing full exploitation of the remaining stored energy [4].

The proposed target-mean SoC controller addresses this limitation by dynamically selecting legal constant-bus configurations that preferentially load cells above the pack-average SoC while allowing lower-SoC cells to rest or be bypassed. As a result, cell depletion is coordinated across the pack, delaying early cutoff and enabling the system to approach the theoretical maximum utilization imposed by the 10% SoC cutoff, while maintaining the required constant-3S bus throughout operation.

Reconfigurable battery packs offer an alternative by selectively altering cell participation under load rather than transferring charge between cells [5–8].

Adaptive reconfiguration has also been investigated as a means of extending battery-system operation by selecting configurations according to real-time load requirements and cell states [9].

Recent work has also explored optimization-based reconfiguration strategies for improving battery-pack utilization. Liu et al. [10] formulated battery-pack reconfiguration as a weighted directed-graph path-planning problem using cell SoC and relay-loss weighting to select high-energy, low-loss discharge paths. Their results showed that path-planning-based reconfiguration can improve pack consistency, reduce relay-related losses, and approach the theoretical utilization limit. However, such approaches require explicit graph-based path optimization, whereas the present work investigates whether a simpler target-mean SoC controller can achieve high utilization using only real-time SoC measurements and a safety-pruned legal action set.

Previous work also investigated self-reconfigurable battery structures that maintain output-voltage stability without requiring an additional DC–DC converter. Ji et al. [11] proposed a self-reconfigurable topology in which higher-SoC cells are preferentially discharged while lower-SoC cells are bypassed, allowing the pack to maintain voltage within a target range and improve capacity utilization. However, such approaches are topology-specific and do not directly address the problem of selecting from among a safety-pruned legal action set under a fixed constant-bus constraint.

Conventional balancing methods are most effective when cells are well matched in capacity and internal resistance [12,13]. In degraded second-life packs, however, capacity mismatch and resistance mismatch weaken this behavior such that equal SoC cannot be assumed to imply equal usable capacity [4,14,15].

Although SoC-based utilization metrics are convenient, they may obscure residual mismatch in absolute remaining capacity when the true objective is runtime improvement [14,15].

This paper investigates whether a simple target-mean SoC controller can provide a better overall pack-level outcome than both parallel-based SoC balancing and mean-based capacity balancing in a heterogeneous constant-3S reconfigurable battery system.

To this end, a target-mean controller is proposed for a constant-3S reconfigurable battery pack. Using only real-time cell SoC measurements and a permissible set of

switching actions, the controller selects the configuration that minimizes deviation from the pack-average SoC while preferentially loading cells with above-average SoC. The approach therefore avoids reliance on active balancing hardware, explicit capacity estimation, and offline optimization during operation.

Unlike conventional mean-based balancing or topology-specific high-SoC selection, the proposed method evaluates the predicted full-pack SoC distribution for every hardware-admissible constant-3S action. Offline pruning ensures electrical admissibility, while the online controller performs lightweight target-mean optimization using only cell SoC feedback. The contribution therefore lies in combining safety-constrained global action evaluation, simple real-time decision logic, comparison against capacity and parallel-based baselines, and hardware validation.

The main contributions of this paper are as follows:

1. Demonstrating that equal state of charge does not necessarily imply equal usable capacity in heterogeneous second-life cells.
2. Proposing a target-mean controller for a constant-3S reconfigurable battery pack based solely on real-time SoC measurements and permissible switching actions.
3. Comparing the proposed controller with mAh-based and parallel-balancing benchmark strategies.
4. Demonstrating through simulation that the proposed controller attains the cutoff-defined 90% theoretical utilization limit and delivers improved runtime under heterogeneous cell conditions.
5. Showing that the proposed controller achieves the best overall compromise between runtime, utilization, balancing quality, and switching activity while reducing switching burden by up to 11.75% relative to the compared methods.
6. Quantifying the controller's computational burden, with $O(N_a N_c)$ per-update complexity, 132 admissible actions, and 660 mask-level cell evaluations per control step, supported by generated-code timing and memory assessment.
7. Experimentally validating the proposed controller on a five-cell hardware prototype, where the fixed-load test achieved 89.12% utilization. A stepped-load test further demonstrated robustness, achieving 88.19% utilization equivalent to 97.99% of the cutoff-defined theoretical limit.

2. Problem Formulation and Motivation

2.1. Effect of Second-Life Cell Mismatch on Balancing

Second-life lithium-ion cells exhibit increased cell-to-cell variability due to non-uniform aging histories, operating temperatures, and degradation rates [1,2,4,14]. In practice, this leads primarily to mismatch in both available capacity and internal resistance, which alters the balancing behavior of the pack under load.

For well-matched cells with similar capacities and internal resistances, balancing through parallel cell operation is generally effective since the cells tend to converge toward a common SoC. In heterogeneous second-life cells, however, this behavior becomes weaker. Wei et al. [14] showed that parallel cells with different capacities can diverge during free discharge, with SoC deviation increasing as the cells discharge, which motivates the need for reconfiguration-based equalization.

Internal-resistance mismatch is particularly important in parallel-connected cells because unequal current sharing can accelerate non-uniform aging and affect pack lifetime [16].

Capacity and internal resistance are commonly assessed during second-life battery repurposing because they directly reflect the usable energy and power capability of reused cells. Braco et al. [2] experimentally demonstrated this using a large second-life

dataset consisting of 506 cells, 203 modules, and 3 battery packs from retired Nissan Leaf vehicles, showing that both capacity and resistance estimation are central to second-life battery evaluation.

Under capacity mismatch, active balancing does not necessarily drive the cells to the same final SoC, and the more significant limitation is that the cells remain mismatched in terms of absolute remaining capacity. Consequently, even if the SoC spread is reduced, the cells do not reach an equivalent usable state. Under internal resistance mismatch, unequal current sharing further distorts the balancing process, such that the cells may also fail to converge to the same final SoC and can remain separated by several percentage points at the end of balancing. This effect becomes increasingly relevant in second-life packs, where resistance growth is a natural consequence of degradation.

Thus, SoC convergence alone cannot be assumed to imply equal remaining absolute capacity in second-life packs.

2.2. SoC and Usable Capacity Are Not Equivalent

State of charge is a convenient normalized measure of relative cell depletion, but it does not directly represent the same quantity of remaining charge in cells with different capacities. For a cell with nominal or available capacity Q_i, the remaining absolute charge can be expressed as:

$$q_i = SoC_i Q_i \tag{1}$$

where q_i is the remaining capacity in absolute units and SoC_i is the normalized state of charge of the i-th cell. For mismatched cells, equal values of SoC_i therefore correspond to different values of q_i. As a result, two cells may appear balanced in SoC while still containing different amounts of usable charge.

Although SoC-based control does not eliminate absolute-capacity mismatch, it may still be effective when the objective is runtime improvement.

2.3. Limitation of SoC-Based Utilization Metrics and Definition of Maximum Utilization

State of charge is commonly used as a normalized measure of cell depletion, and many utilization-oriented battery studies evaluate performance using a fixed end-of-discharge threshold, such as a 10% SoC cutoff [14,15]. Under this type of cutoff-defined evaluation, the maximum extractable fraction of a fully charged cell is limited to 90% because the final 10% of cell capacity is intentionally preserved. Therefore, for full-initial-SoC cases in this work, the theoretical maximum capacity utilization is defined as:

$$\eta_{max} = (1 - SoC_{cutoff}) \times 100\% \tag{2}$$

where $SoC_{cutoff} = 0.10$. This gives:

$$\eta_{max} = (1 - 0.1) \times 100 = 90\% \tag{3}$$

In this paper, the term maximum utilization refers specifically to this cutoff-defined maximum utilization under the imposed lower SoC limit. It should not be interpreted as the true physical or electrochemical maximum utilization of the cells since deeper discharge below the selected cutoff is intentionally excluded for safety and practical battery-management reasons. Thus, the reported 90% value represents the theoretical upper bound imposed by the selected 10% per-cell SoC cutoff criterion.

This distinction is especially important for heterogeneous second-life cells. Since SoC is a normalized quantity, equal SoC values do not necessarily correspond to equal absolute remaining capacity when cells have different available capacities [14,15]. A controller may therefore produce apparently balanced SoC trajectories while the cells still contain different absolute amounts of remaining charge. Conversely, a capacity-based controller

may balance absolute remaining capacity but still terminate earlier if one cell reaches the SoC cutoff before the others.

For this reason, runtime and capacity utilization are treated as related but distinct performance metrics in this study. Runtime measures the discharge duration until the first cell reaches the imposed 10% SoC cutoff, whereas capacity utilization measures the fraction of initial available pack capacity extracted before this cutoff. The proposed target-mean SoC controller is therefore evaluated according to its ability to coordinate cell depletion relative to the cutoff condition, delay early cell cutoff, and approach the cutoff-defined maximum utilization under the tested heterogeneous pack conditions.

2.4. Control Objective and Problem Statement

This work is built upon a previously developed five-cell, 19-switch reconfigurable battery framework [17], which provides a sufficiently flexible action space for selective cell participation, bypass operation, and constant-3S bus control. A newly drawn schematic of the topology used in the present study is shown in Figure 1. Unlike hierarchical multiscale reconfiguration strategies, the present work considers direct online selection among legal constant-3S switching actions at the individual-cell participation level.

In this work, the term constant-bus constraint refers to maintaining a fixed three-cell series output path, rather than regulating the bus voltage to an exactly constant numerical value. The motivation follows previous self-reconfigurable battery work, where reconfiguration was used to keep the battery-pack voltage within a suitable operating range and reduce the need for large voltage-conversion stages [11]. In the present five-cell prototype, enforcing constant-3S operation constrains the load-side voltage to a predictable 3S operating window, approximately 8–10.5 V under the tested LFP discharge conditions. This is more practical for the load or downstream converter than arbitrary series-cell-count operation, where the output voltage could vary substantially as cells are inserted or bypassed.

The predictability of the constant-3S bus is also supported by the cell SoC–OCV relationship. Because the controller coordinates cell depletion and selects cells from the same chemistry and voltage range, the three cells participating in the series path have broadly similar terminal-voltage characteristics during operation. The bus voltage is therefore mainly the sum of three comparable cell voltages, with the remaining variation caused by SoC differences, internal resistances, and load current. Thus, “constant-bus” in this paper should be interpreted as a fixed 3S bus constraint with bounded voltage variation, not active DC-DC-style voltage regulation.

The controller is not permitted to search arbitrary 19-switch combinations online. Instead, it operates within a precomputed legal action space generated by offline safety pruning. At a high level, the pruning removes switch states that violate the constant-3S bus requirement, create invalid end-node selection, remove the required ground or conduction path, enable conflicting forward/reverse/current-path switches, produce short-circuit-like behavior, exceed allowable load or cell-current limits, or cause reverse cell current during discharge. The objective of the online controller is therefore to optimize SoC coordination only among hardware-admissible switching states.

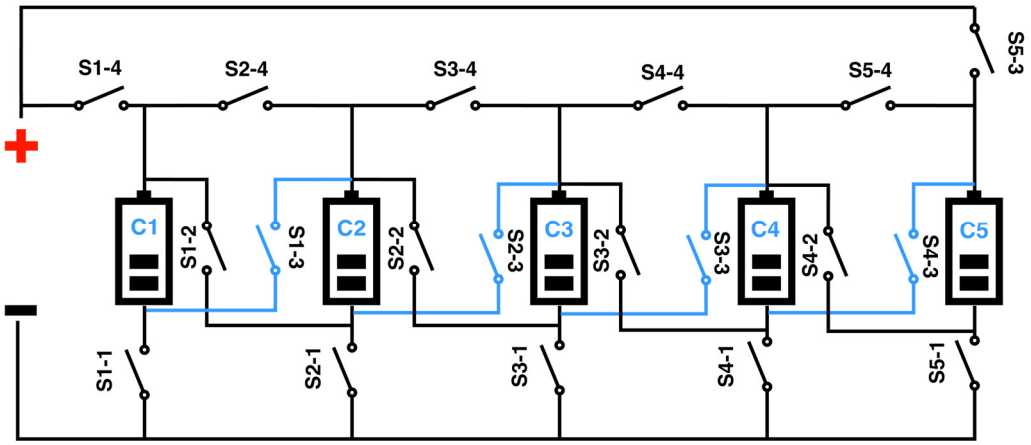

Figure 1. Five-cell, 19-switch reconfigurable battery framework supporting constant-3S operation, based on the topology introduced in [17].

This paper considers a constant-3S reconfigurable battery pack operated under a legal set of switching actions. At each control instant, the controller selects one admissible configuration from the action set such that exactly three cells are actively connected in series, while the remaining cells are bypassed or paralleled. Let the cell SoC vector at time step k be defined as:

$$s(k) = [s_1(k)s_2(k)s_3(k)s_4(k)s_5(k)]^T \quad (4)$$

And let the corresponding pack-average SoC be:

$$\mu(k) = \frac{1}{5}\sum_{i=1}^{5} s_i(k) \quad (5)$$

The control problem is to select, in real time, the legal switching action that

- reduces deviation of the cell SoCs from the pack-average value,
- preferentially loads cells above the mean SoC,
- improves total pack runtime,
- and achieves the theoretical maximum utilization limit defined by the SoC cutoff criterion.

Unlike SoH-aware or capacity-aware approaches, the controller developed in this work does not require explicit estimation of capacity fade, internal resistance, or cell health states [4,14,15]. Instead, it uses only real-time SoC measurements and the legal action space of the reconfigurable pack. In the hardware implementation, experimentally measured cell capacities are used only within the Coulomb-counting SoC estimator and not used by the proposed target-mean controller as capacity-based inputs. The aim is to determine whether such a simple SoC-based strategy can be sufficient for both runtime-oriented control and theoretical maximum utilization in mismatched second-life packs.

Prior SoH-aware reconfiguration studies have shown that grouping cells of similar health within the same series string can improve delivered pack capacity in heterogeneous battery packs, since the usable capacity of a series string is constrained by its weakest cell [4]. In contrast, the present work investigates whether comparable pack-level benefits can be achieved without explicit SoH estimation. Specifically, the aim is to determine whether, within a constant-3S reconfigurable framework considered here, a simple target-mean SoC controller based only on real-time SoC measurements can provide the best overall compromise between runtime, utilization, residual imbalance, and switching activity. The results indicate that, under the evaluated mismatch scenarios, explicit SoH-aware control is not required to obtain strong pack-level performance.

3. Target-Mean Controller

3.1. Legal Action Space Construction

The five-cell, 19-switch reconfigurable battery framework admits 2^{19} switch combinations, many of which are invalid, unsafe, or incompatible with the constant-3S bus constraint [18].

This is consistent with prior reviews of reconfigurable battery systems, which identify safe and optimal utilization, hardware overhead, switching losses, and scalability as central challenges in practical RBS deployment [5,6,19].

To reduce this space, a two-step offline pruning procedure was applied, as summarized in Algorithm 1. Clearly invalid and short-circuit-prone combinations were first removed using logical filtering, after which the remaining candidates were screened using a short simulation-based safety check.

Algorithm 1: Two-Step Legal Action Space Pruning and *SeriesMask* Generation

Input: Full 19-switch action space, logical pruning rules, simulation safety criteria, constant-3S operating constraint

Output: Legal action set $\mathcal{A}$, *SeriesMask*

1. Initialize the full candidate action set $\mathcal{A}_{\text{cand}} \leftarrow \{0,1,\dots,2^{19}-1\}$
2. Apply logical pre-pruning to remove clearly invalid or unsafe combinations, including:
   a) inconsistent end-node selection
   b) missing ground paths
   c) missing conduction paths
   d) forbidden simultaneous switch activations
3. Set the surviving actions as the pre-pruned candidate set $\mathcal{A}_{pre} \leftarrow$pre-pruned actions
4. Initialize the legal action set $\mathcal{A} \leftarrow \emptyset$
5. **For each** candidate action $a_j \in \mathcal{A}_{pre}$
6. Run a short simulation for action a_j
7. Measure the resulting bus voltage and cell/load currents
8. Reject action a_j if any of the following conditions is satisfied:
   a) $V_{bus} \notin [8.5, 11]\,V$;
   b) short-circuit-like behavior is detected or load current is below $0.2\ A$;
   c) circulating current exceeds $5\ A$ or load current exceeds $10\ A$;
   d) reverse cell current exceeds $3\ A$;
   e) any cell current exceeds $5\ A$.
9. **If** action a_j satisfies all safety and constant-3S requirements:
10. Add a_j to the legal action set $\mathcal{A} \leftarrow \mathcal{A} \cup \{a_j\}$
11. Generate the corresponding binary *SeriesMask* row $\boldsymbol{m}_j = [m_{j,1}\ m_{j,2}\ m_{j,3}\ m_{j,4}\ m_{j,5}]$
12. Retain only masks satisfying $\sum_{i=1}^{5} m_{j,i} = 3$
13. **End for**
14. Form the *SeriesMask* matrix by stacking the retained mask rows corresponding to all legal actions in $\mathcal{A}$
15. Return $\mathcal{A}$ and *SeriesMask*

The simulation-based pruning stage used the same safety-pruned action-space construction previously developed for the five-cell, 19-switch constant-3S topology in [17]. Each pre-pruned candidate was evaluated over a short $T = 0.25\,s$ Simscape simulation window with a maximum solver step of 0.005 s. Candidate actions were retained only if they satisfied the three-cell-equivalent bus-voltage requirement and did not trigger any electrical safety conditions. The safety thresholds used in the pruning process were a short-circuit voltage threshold of $V_e = 0.05\,V$, a near-zero load-current threshold of $I_{small} = 0.2\,A$, a circulating-current threshold of $I_{circ} = 5\,A$, a maximum load-current limit of $I_{load,max} = 10\,A$, a reverse-cell-current limit of $I_{rev,max} = 3\,A$, and a maximum per-cell-current limit of $I_{cell,max} = 5\,A$. These thresholds were selected as conservative offline screening limits informed by $LiFePO_4$ cell operating limits and the required three-cell-equivalent bus-voltage range. These values were used only for offline action-space pruning; the online target-mean controller then evaluates only the retained legal actions.

Applying Algorithm 1 reduced the full 2^{19} switching space to 132 safe legal actions. These retained actions form the controller action set, denoted by (6). The use of a safety-pruned legal action space ensures that the online controller evaluates only admissible switching states, thereby improving both computational efficiency and operational safety.

$$\mathcal{A} = \{a_1, a_2, \ldots, a_N\}, N_a = 132 \quad (6)$$

Although all 132 retained actions satisfy the constant-3S participation requirement, they correspond to 10 unique three-cell participation masks, equivalent to the number of ways of selecting three active cells from five. Multiple switching configurations can therefore realize the same active-cell set because of the topology routing, bypass, and parallel-connection possibilities. The controller evaluates the safe action library using the associated *SeriesMask* representation, so the online decision remains computationally lightweight while still preserving hardware-feasible switching options.

Where multiple legal switching configurations produce the same three-cell participation mask, they receive identical SoC-based costs. Ties are resolved deterministically by selecting the first encountered minimum-cost action in the stored legal-action ordering.

The per-step computational complexity of the online target-mean evaluation is $O(N_a N_c)$, where N_a is the number of admissible actions and N_c is the number of cells. For the evaluated prototype, $N_a = 132$ and $N_c = 5$, corresponding to 660 mask-level cell evaluations per control step. This supports real-time implementation at the 1 s hardware update interval used in the experimental validation.

The retained legal action set forms the basis of the proposed target-mean controller. At each control step, the controller evaluates these admissible actions using the measured cell SoCs and selects the action with minimum cost. The overall real-time decision process is summarized in Figure 2.

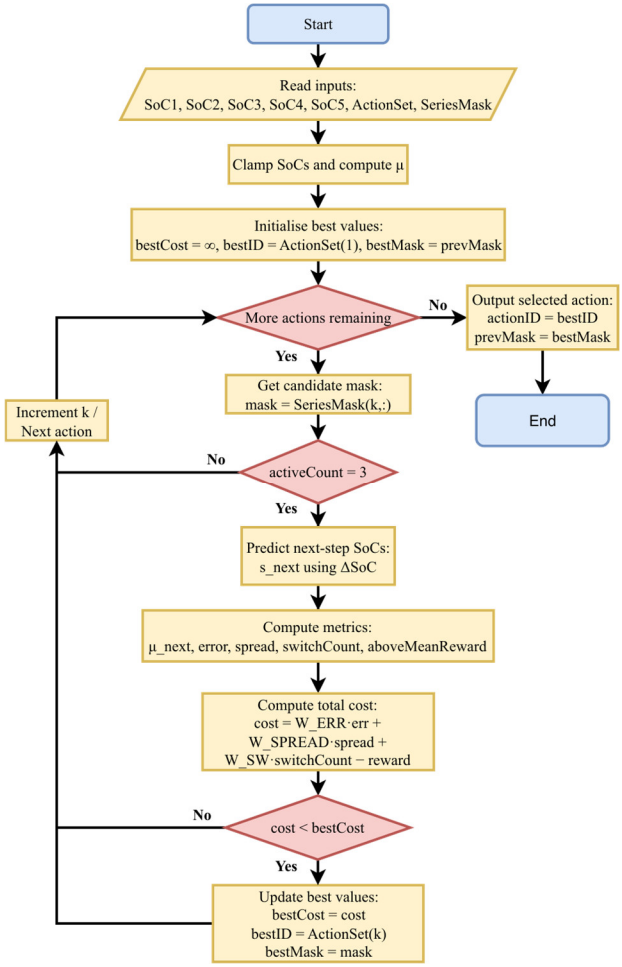

Figure 2. Flowchart of the proposed target-mean controller.

3.2. Series Participation Mask and Candidate State Prediction

For each retained legal action $a_j \in \mathcal{A}$, the corresponding constant-3S cell participation is represented by a binary series mask:

$$\boldsymbol{m}_j = [m_{j,1}\ m_{j,2}\ m_{j,3}\ m_{j,4}\ m_{j,5}] \tag{7}$$

where:

$$m_{j,i} = \begin{cases} 1, & \text{if cell } i \text{ is active under action } a_j \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

and

$$\sum_{i=1}^{5} m_{j,i} = 3, \forall a_j \in \mathcal{A}. \tag{9}$$

Let the measured cell SoC vector at time step k be:

$$\boldsymbol{s}(k) = [s_1(k)\ s_2(k)\ s_3(k)\ s_4(k)\ s_5(k)]^T \tag{10}$$

With corresponding pack-average SoC:

$$\mu(k) = \frac{1}{5}\sum_{i=1}^{5} s_i\,(k) \tag{11}$$

For each candidate action a_j, the one-step predicted SoC of cell i is:

$$\hat{s}_{i,j}(k+1) = sat,\left(s_i(k) - m_{j,i}\Delta_{SoC}\right) \tag{12}$$

where $sat(\cdot)$ denotes saturation to $[0,1]$. The corresponding predicted pack-average SoC is:

$$\hat{\mu}_j(k+1) = \frac{1}{5}\sum_{i=1}^{5} \hat{s}_{i,j}\,(k+1). \tag{13}$$

3.3. Target-Mean Cost Function

Each legal action is evaluated using a cost function that penalizes deviation from the predicted pack-average SoC, penalizes predicted SoC spread, optionally penalizes switching changes relative to the previously selected mask, and rewards the selection of cells that are currently above the pack-average SoC.

$$J_{err,j}(k) = \sum_{i=1}^{5} (\hat{s}_{i,j}\,(k+1) - \hat{\mu}_j(k+1))^2 \tag{14}$$

$$J_{spread,j}(k) = \max_i\ \hat{s}_{i,j}(k+1) - \min_i\ \hat{s}_{i,j}(k+1) \tag{15}$$

$$J_{sw,j}(k) = \sum_{i=1}^{5} |\, m_{j,i} - m_{prev,i}\,| \tag{16}$$

$$R_{mean,j}(k) = \sum_{i=1}^{5} m_{j,i}\, max\,(s_i(k) - \mu(k), 0) \tag{17}$$

The total cost is:

$$J_j(k) = W_{err}J_{err,j}(k) + W_{spread}J_{spread,j}(k) + W_{sw}J_{sw,j}(k) - R_{mean,j}(k) \tag{18}$$

In this work,

$$W_{err} = 1.0, \qquad W_{spread} = 0.5, \qquad W_{sw} = 0.0 \tag{19}$$

With

$$\Delta_{SoC} = 0.002 \tag{20}$$

Although a switching-count penalty term is included in the controller formulation for generality, it was set to zero in this study ($W_{sw} = 0$). Consequently, the observed reduction in switching activity emerged from the action-selection behavior itself rather than from explicit penalization.

To assess the contribution of the individual cost terms, ablation testing was conducted by evaluating reduced cost-function formulations against the full controller under the mismatch scenarios described in Section 4. The results showed that the mean-error term is the primary driver of pack-level performance. Removing the spread and above-mean reward terms produced no measurable degradation in utilization, runtime, or final SoC spread under the evaluated conditions. This indicates that direct targeting of the pack-average SoC is the dominant mechanism behind the proposed controller's performance. The spread and above-mean terms are therefore retained in the full formulation as secondary refinements, providing additional generality for operating conditions beyond those evaluated here. Importantly, this result supports the broader conclusion that effective reconfiguration control for heterogeneous second-life battery systems does not necessarily require complex multi-objective formulations; instead, a simple target-mean SoC controller can be sufficient to reach the cutoff-defined theoretical utilization limit under the tested scenarios.

3.4. Real-Time Action Selection

At each control instant, all legal actions in $\mathcal{A}$ are evaluated using the predicted SoC update and the cost function defined above. The selected action is the minimum-cost action:

$$a^{\star}(k) = arg \min_{a_j \in \mathcal{A}} J_j(k) \quad (21)$$

and its corresponding action ID is returned to the controller output. The associated mask is then stored as the new previous mask for the next control step.

4. Simulation Setup and Comparison Methods

4.1. Cell Model and Mismatch Conditions

Each cell was represented using a standard lookup-table battery model, in which open-circuit voltage and terminal resistance were defined as functions of state of charge. This modeling approach is commonly used for controller-oriented battery simulation because it captures the principal dependence of terminal behavior on SoC without requiring a full electrochemical model.

The proposed controller was evaluated in simulation using a five-cell reconfigurable battery pack operated under the constant-3S constraint described in Section 3. The study considered mismatched-cell scenarios representative of second-life battery behavior, with differences introduced in cell capacity and internal resistance. A SoC-dependent internal resistance trend, typically increasing at low and high SoC, has been reported in lithium-ion battery modeling studies and was adopted here to represent resistance variance under degraded or heterogeneous operating conditions [20]. These mismatch conditions were selected to assess the extent to which the controller could maintain coordinated discharge, extend runtime, and achieve high utilization despite degraded cell consistency.

All controllers were evaluated using the same pack framework, action-space constraints, and end-of-discharge criterion in order to ensure a fair comparison. Capacity utilization was assessed using a 10% SoC cutoff, giving a theoretical maximum utilization limit of 90%.

4.2. Comparison Controllers

To assess whether explicit capacity-aware control is necessary for runtime-oriented operation, the proposed target-mean SoC controller was compared against two reference methods. The evaluated controllers are summarized in Table 1.

Table 1. Comparison of evaluated controllers.

Controller	Decision Variable	Action-Selection Method
Target-mean SoC	SoC	Minimum-cost action relative to pack-average SoC
Target-mean mAh	Remaining absolute Capacity	Minimum-cost action relative to mean remaining capacity
Parallel Balancing	SoC imbalance	Parallel-capable midpoint heuristic

The target-mean mAh-based controller followed the same overall action-selection structure as the proposed method but replaced SoC with remaining absolute capacity as the balancing variable [14,15]. Candidate actions were therefore evaluated according to their predicted effect on the mean remaining capacity rather than the mean SoC.

The parallel-balancing baseline operated on the subset of actions that permit parallel cell interaction [11,12]. At each control step, it identified cells above and below the pack-average SoC and selected a parallel-capable action intended to bring the associated cells toward the midpoint. Unlike the proposed controller, it did not evaluate the full legal action set using an explicit global cost function.

Across the simulation and hardware comparisons, all controllers used identical conditions, decision intervals, end-of-discharge criteria, and applicable pack constraints. The hardware tests additionally used the same measurement and SoC-estimation procedures,

while both target-mean controllers used the same legal action library and deterministic action ordering.

4.3. Performance Metrics

Controller performance was assessed using the metrics listed in Table 2. These were selected to distinguish between balancing quality and practical pack-level performance under second-life mismatch.

Runtime is defined as the elapsed discharge time from the start of operation until the first cell reaches the 10% SoC cutoff. Capacity utilization is calculated over the same interval as the fraction of initial available pack capacity extracted before this cutoff. Therefore, both runtime and capacity utilization are evaluated under the same practical end-of-discharge condition.

Table 2. Performance metrics.

Metric	Description	Purpose
Runtime	Total discharge duration until the 10% SoC cutoff	Primary practical performance measure
Capacity utilization	Fraction of usable pack capacity delivered before the 10% SoC cutoff	Measures proximity to the 90% theoretical limit
Relative SoC spread	Difference between maximum and minimum cell SoC during discharge	Indicates balancing effectiveness

Capacity utilization was defined as the percentage of initial available pack capacity released before the 10% SoC cutoff. For cases where all cells began from full charge, this gives a theoretical upper limit of 90%. For initial pack capacity C_{init} and remaining pack capacity at cutoff C_{left}, the utilization is:

$$P = \frac{C_{init} - C_{left}}{C_{init}} \times 100\% \quad (22)$$

For the unequal-initial-SoC case, the 10% cutoff is enforced at the individual-cell level rather than as 10% of the initial available pack charge. Therefore, the remaining pack capacity at cutoff is obtained by summing the residual 10% capacity of each cell. For the cell capacities $[1400,1100,1600,1320,900]$ mAh, this gives a cutoff residual of 632 mAh. Since the cells begin from unequal initial SoCs, the initial available pack charge is 5446 mAh rather than the full nominal pack capacity, and the resulting case-specific maximum utilization is therefore $(5446 - 632)/5446 \times 100 = 88.40\%$, rather than 90%.

5. Simulation Results

5.1. Results Under Internal-Resistance Mismatch

Figure 3 compares the balancing behavior of the evaluated controllers under internal-resistance mismatch, while keeping the initial cell SoC and capacity identical. In this case, the mismatch arises purely from resistance variation, allowing the effect of unequal current sharing on balancing behavior to be isolated. The proposed target-mean SoC controller is compared against the target-mean mAh-based controller and the parallel-balancing controller using the same constant-3S pack framework and 10% SoC cutoff criterion.

The results show that internal-resistance mismatch alone is sufficient to degrade balancing performance, even when the cells begin with identical SoC and capacity. Under resistance mismatch alone, the target-mean SoC and target-mean mAh controllers produced identical runtime, utilization, and final SoC-spread results because the cell

capacities were equal. The parallel-based controller retained a small residual SoC spread, terminated earlier, and extracted less capacity.

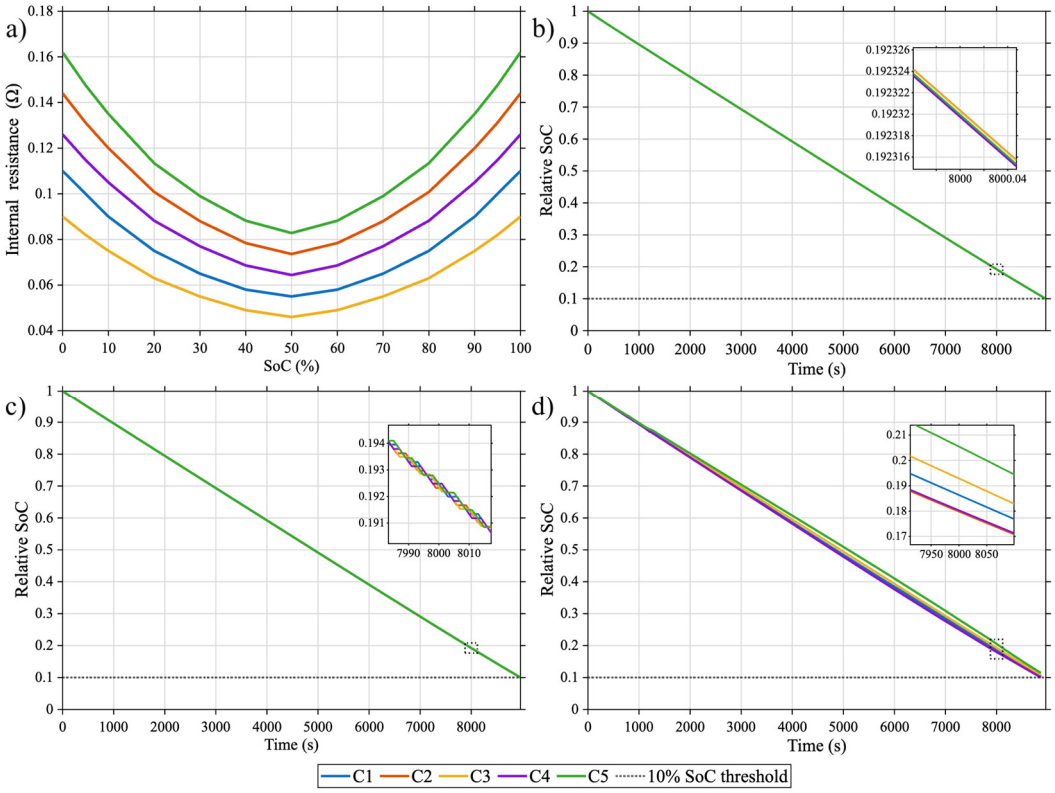

Figure 3. Controller comparison under internal-resistance mismatch: (**a**) cell internal-resistance profiles, (**b**) target-mean SoC-based controller, (**c**) target-mean mAh-based controller, and (**d**) parallel-balancing controller.

5.2. Results Under Capacity Mismatch

Figure 4 compares the balancing behavior of the evaluated controllers under capacity mismatch, while keeping the initial cell SoC and internal resistance identical. In this case, only the available cell capacities are varied, with nominal values of [1400, 1100, 1600, 1320, 900] mAh, where 1600 mAh represents the ideal cell capacity. The proposed target-mean SoC controller is again compared against the target-mean mAh-based controller and the parallel-balancing baseline using the same constant-3S pack framework and 10% SoC cut-off criterion.

The results indicate that balancing absolute remaining capacity does not necessarily yield the best pack-level outcome. Instead, the proposed SoC-based controller provides the best compromise between coordinated discharge, runtime, and utilization. In particular, it achieves the most coordinated discharge behavior and reaches the 90% theoretical utilization limit. By comparison, the parallel-balancing baseline exhibits substantial mid-discharge divergence before the trajectories begin to reconverge, while the mAh-based controller performs worst in SoC-balancing terms despite driving the cells toward similar final absolute remaining capacities. These results further support the view that balancing absolute capacity alone does not necessarily improve practical pack runtime, and that SoC-based reconfiguration control remains more effective for runtime-oriented operation under second-life mismatch.

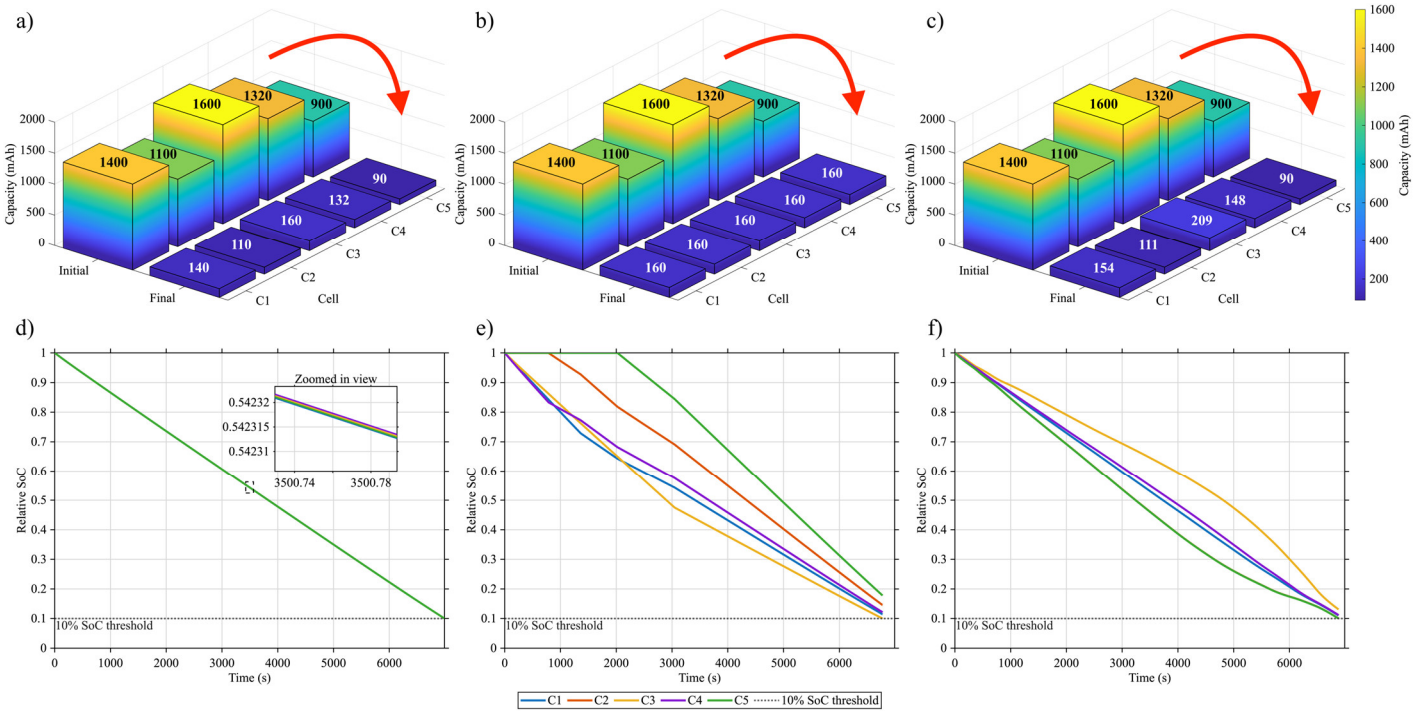

Figure 4. Controller comparison under capacity mismatch: (**a**) target-mean SoC-based controller, (**b**) target-mean mAh-based controller, and (**c**) parallel-balancing controller, with (**d**–**f**) showing the corresponding relative SoC spreads.

5.3. Switching Activity

Figure 5 compares the topology-aligned per-switch turn-on counts of the evaluated controllers. The proposed target-mean SoC controller exhibits the lowest overall switching activity by up to 11.75%, while still achieving the best runtime and utilization performance. This is practically important because fewer switching events reduce switching burden on the hardware and may help reduce switching losses. By contrast, the mAh-based and parallel-based controllers require more frequent switch operation to achieve their behavior. These results suggest that the proposed controller improves pack-level performance without increasing switching effort, which strengthens its practical suitability for reconfigurable battery implementation.

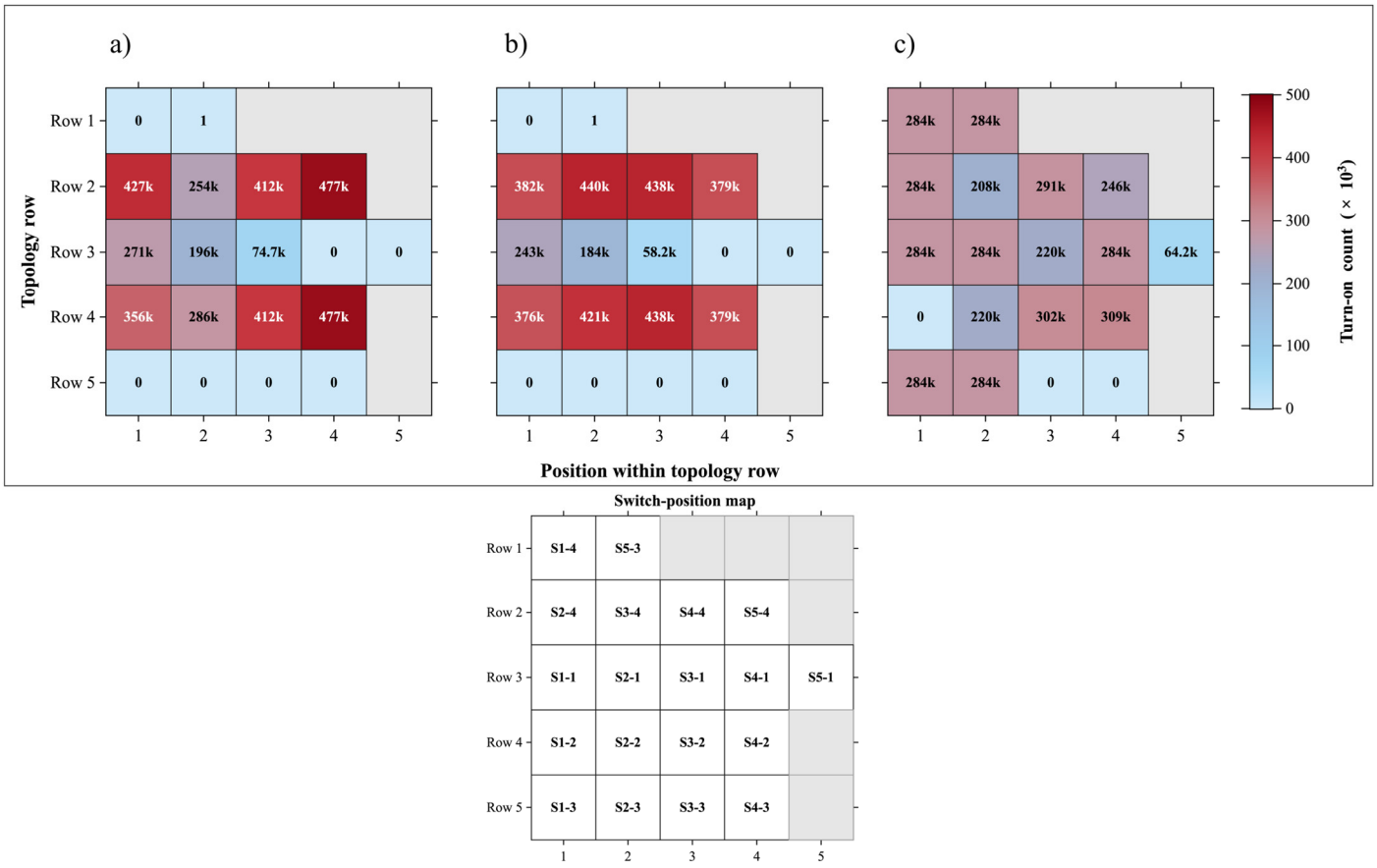

Figure 5. Topology-aligned heatmaps of per-switch turn-on counts for (**a**) target-mean SoC, (**b**) target-mean mAh, and (**c**) parallel-based control. The proposed target-mean SoC controller exhibits the lowest overall switching activity, indicating the lowest switching burden among the evaluated methods.

5.4. Summary of Simulation Results

Table 3 summarizes the controller performance under the individual mismatch cases. In both internal-resistance and capacity-mismatch scenarios, the proposed target-mean SoC controller achieved the best overall performance, combining the highest utilization with the smallest residual SoC spread. Table 4 summarizes the combined resistance- and capacity-mismatch case under a 10 Ω load. In this case, the proposed controller again provided the best overall compromise, achieving the highest utilization among the evaluated methods while also requiring the lowest switching activity. Taken together, the results indicate that the target-mean SoC controller provides the best compromise between runtime, utilization, balancing quality, and switching burden.

This confirms that, for the mismatch scenarios considered here, the proposed target-mean SoC strategy provides the best overall compromise between runtime, utilization, and residual imbalance at the 10% per-cell SoC cutoff, compared with both parallel-based SoC balancing and mean-based capacity balancing.

Table 3. Summary of controller performance under individual mismatch conditions.

Load (Ω)	Scenario	Controller	Runtime to 10% SoC Cutoff (s)	Utilization (%)	Capacity Remaining (mAh)	SoC Spread at 10% Cutoff (p.p.)
10	Internal resistance mismatch	Target-mean SoC	8964.15	90.00	800.004	0.000

10	Internal resistance mismatch	Target-mean mAh	8964.15	90.00	800.004	0.000
10	Internal resistance mismatch	Parallel-based	8870.00	89.48	841.455	1.431
10	Capacity mismatch	Target-mean SoC	6991.91	90.00	632.006	0.000
10	Capacity mismatch	Target-mean mAh	6774.13	87.34	799.994	7.780
10	Capacity mismatch	Parallel-based	6880.56	88.74	711.597	3.073

This is in agreement with Wei et al. [14], who reported SoC divergence during free discharge of parallel cells with different capacities. In the present study, the parallel-based controller likewise retained residual imbalance, with a SoC spread of 1.431 p.p. under internal-resistance mismatch and 3.073 p.p. under capacity mismatch at the 10% cutoff. These results indicate that parallel balancing alone does not necessarily ensure full convergence in heterogeneous packs.

In the combined SoC, resistance, and capacity mismatch case, the proposed target-mean SoC controller reached the theoretical residual capacity limit of approximately 632 mAh at the 10% cutoff, corresponding to 88.40% utilization. This represents the case-specific theoretical maximum under the unequal initial-SoC condition and 10% cutoff. By contrast, the mAh-based controller terminated with 800.00 mAh remaining, corresponding to 85.31% utilization. The proposed controller also increased runtime from 5795.56 s to 6017.76 s, an improvement of 222.20 s, or 3.83%. This confirms that coordinating cell depletion around the pack-average SoC improves both cutoff-defined capacity utilization and practical runtime compared with balancing remaining absolute capacity alone.

Table 4. Summary of controller performance under combined resistance, capacity, and SoC mismatch case ($SoC_{1,\mathrm{init}} = 0.99$, $SoC_{2,\mathrm{init}} = 0.90$, $SoC_{3,\mathrm{init}} = 0.70$, $SoC_{4,\mathrm{init}} = 0.85$, $SoC_{5,\mathrm{init}} = 0.92$) with a 10 Ω load, where the theoretical limit of maximum utilization is 88.40%.

Load (Ω)	Scenario	Controller	Runtime (s)	Total Switch-Ons up to 10% SoC	Normalized Switching Activity (Switch-Ons/s)	Utilization (%)	Capacity Remaining (mAh)	SoC Spread at 10% Cutoff (p.p.)
10	Combined SoC, Resistance and Capacity mismatch	Target-mean SoC	6017.76	3,644,356	605.60	88.40	632.00	0.000
10	Combined SoC, Resistance and Capacity mismatch	Target-mean mAh	5795.56	3,740,478	645.40	85.31	800.00	7.782
10	Combined SoC, Resistance and Capacity mismatch	Parallel-based	5953.61	4,129,525	693.62	87.78	665.27	1.470

6. Experimental Analysis

6.1. Experimental Setup

For hardware implementation, a deployable real-time model was developed for measurement acquisition, signal conditioning, switch activation, and calibrated Coulomb-counting-based SoC estimation. Measured per-cell currents were integrated using the experimentally characterized capacities in Table 5 to provide the relative SoC feedback required by the host-level target-mean controller.

This estimation layer was executed on the embedded platform and provided the state information required by the host-level target-mean controller.

A hardware prototype of the reconfigurable battery system was developed to support implementation of the proposed control framework on real cells. The experimental setup is shown in Figure 6 and consists of the following:

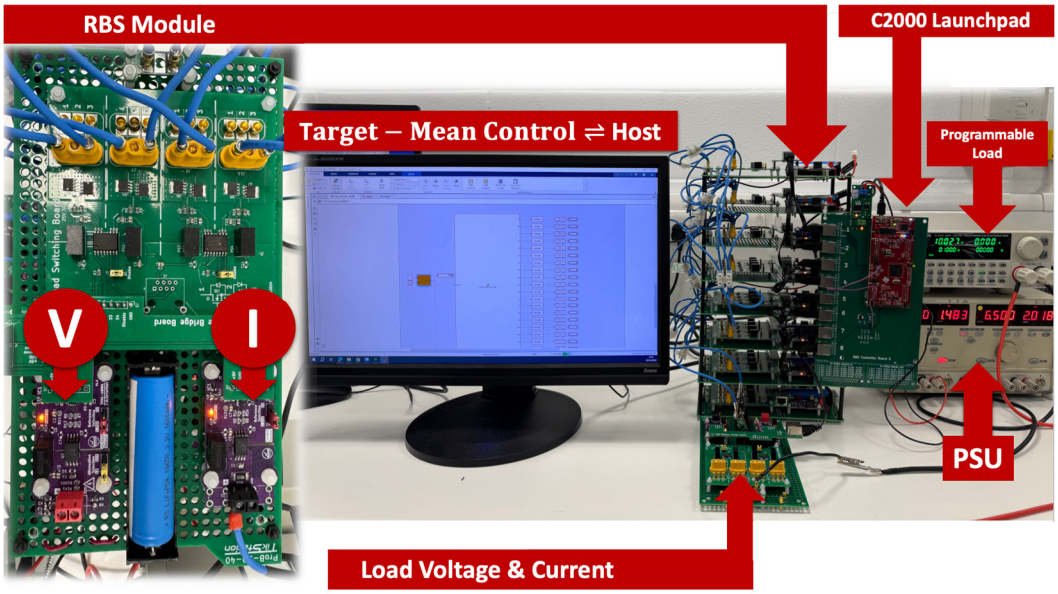

Figure 6. Hardware prototype of the reconfigurable battery system, showing the RBS module, quad switching board, per-cell voltage and current sensing, host PC running the control algorithm, TI C2000 LAUNCHXL-F28379D, programmable load, and supporting power supply.

- Five unbranded commercial LFP cells with measured capacities of 1634 mAh, 1563 mAh, 1820 mAh, 1560 mAh, and 1542 mAh.
- RBS module, used as the main reconfigurable battery system control platform.
- Quad switching board populated with real MOSFETs and gate drivers, designed by TikStation Ltd. (Sheffield, UK), enabling arbitrary battery topologies to be assembled and controlled in real time.
- Custom per-cell voltage and current sensing, used for measurement acquisition and closed-loop monitoring of individual cells.
- TI C2000 LAUNCHXL-F28379D development board (Texas Instruments, Dallas, TX, USA), used for measurement acquisition, Coulomb-counting-based SoC estimation, and switch activation.
- Host PC, used to execute the target-mean controllers and comparison methods, and to communicate with the embedded platform through a serial link.
- Programmable electronic load, TENMA 72-13210 / Multicomp Pro 72-13210 (Premier Farnell UK Ltd., Leeds, UK), used to apply the discharge current.
- External PSU, TTi QL335T (Thurlby Thandar Instruments Ltd., Huntingdon, UK), used to power RBS controller board.

- BK Precision BA6011 battery analyzer (B&K Precision Corp., Yorba Linda, CA, USA), used for cell internal-resistance measurement.
- Varicore VCP4 charger/tester (VariCore/YinQian Electronic Technology Co., Ltd., Shenzhen, China), used for practical capacity testing and charging of cells.

Figure 7 summarizes the real-time implementation architecture. The TI C2000 LAUNCHXL-F28379D acquires the per-cell voltage and current measurements, performs calibrated Coulomb-counting estimation, and applies the selected switching action. The estimated cell SoCs are transmitted to the host PC, where the target-mean controller evaluates the legal action set and returns the selected action ID to the embedded platform for switch actuation.

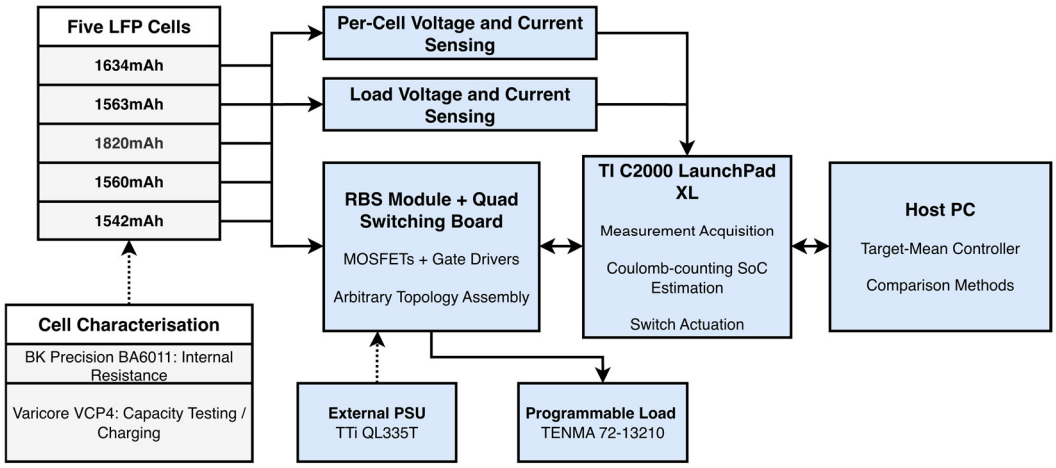

Figure 7. Real-time implementation architecture of the proposed hardware platform, showing measurement acquisition, Coulomb-based SoC estimation, and host-level target-mean control.

Table 5. Cell characterization parameters.

Cell Number	Labeled Capacity (mAh)	Measured Capacity (mAh)	Internal Resistance at 100% SoC (mΩ)
1	1600	1634	35.4
2	1000	1563	90.4
3	1800	1820	46.5
4	1500	1560	94.2
5	1200	1542	56.2

The measured capacities ranged from 1542 mAh to 1820 mAh, giving a maximum spread of 278 mAh (approximately 18.0% relative to the lowest-capacity cell). This mismatch is sufficient to emulate heterogeneous pack conditions for hardware validation.

Capacity characterization showed that several cells did not match their nominal labeled capacities. In particular, cells labeled as 1000 mAh and 1200 mAh exhibited measured capacities substantially above those values, even when measured twice, whereas other cells were closer to their nominal ratings. Although the hardware study used new cells, these findings highlight the importance of experimental cell characterization when creating pack conditions intended to mimic heterogeneous second-life behavior, since nominal labels may not reliably reflect actual usable capacity.

6.2. Hardware Control Update Interval

In simulation, the controller was evaluated with a shorter decision interval to provide a high temporal resolution for switching behavior. For hardware implementation, however, a 1 s supervisory update interval was adopted for all control strategies in order to

allow sufficient time and a more accurate reading for Coulomb-counting-based SoC estimation, signal conditioning, communication, and reliable switching execution on the embedded platform.

A sensitivity comparison of the target-mean controller at 0.005 s and 1 s decision intervals showed that the pack-level behavior was preserved, supporting the use of the slower update rate in hardware and reducing switching activity further without the need for a faster decision rate. The results are summarized in Table 6.

Table 6. Decision interval comparison under combined resistance, capacity, and SoC mismatch case ($SoC_{1,\mathrm{init}} = 0.99$, $SoC_{2,\mathrm{init}} = 0.90$, $SoC_{3,\mathrm{init}} = 0.70$, $SoC_{4,\mathrm{init}} = 0.85$, $SoC_{5,\mathrm{init}} = 0.92$) with a 10 Ω load, where the theoretical limit of maximum utilization is 88.40%.

Decision Interval	Scenario	Controller	Runtime (s)	Total Switch-Ons up to 10% SoC	Utilization (%)	SoC Spread at 10% Cut-off (p.p.)
0.005 s	Combined SoC, Resistance and Capacity mismatch	Target-mean SoC	6017.76	3,644,356	88.40	0.000
1 s	Combined SoC, Resistance and Capacity mismatch	Target-mean SoC	6017.76	18,160	88.39	0.005

Increasing the decision interval from 0.005 s to 1 s reduced switching activity by 99.50% while changing utilization by only 0.01 p.p. This indicates that cell-SoC dynamics are slow relative to the control interval and that most high-frequency action updates do not materially alter the long-term cell-participation allocation.

6.3. Computational Performance and Code-Generation Assessment

The experimental system used host-supervised action selection, while the C2000 performed measurement acquisition, Coulomb-counting SoC estimation, and switch actuation. The target-mean controller was therefore benchmarked post-experimentally using the same 132-action library and 132×5 *SeriesMask* used in the hardware tests, corresponding to 660 mask-level cell evaluations per update.

Across 30,000 evaluations, the MATLAB implementation achieved a median execution time of 33.427 µs, while the generated C achieved 3.825 µs. The generated-C 99th-percentile time was 5. 375 µs, equivalent to 0.000538% of the 1 s control interval. Generated C matched the MATLAB controller for all 10,000 verification cases.

Static analysis reported 115 controller code lines, a cyclomatic complexity of 27, an accumulated stack estimate of 246 bytes, 42 bytes of generated global data, and 1716 bytes of action-library storage. These results quantify software-level computational burden and should not be interpreted as measured C2000 execution time or processor utilization.

The benchmark was conducted on an Apple M4 MacBook Pro (Apple Inc., Cupertino, CA, USA), with 16 GB of unified memory, running macOS Tahoe version 26.3.1 and MATLAB R2025a Update 1.

6.4. Hardware Capacity Utilization

For the hardware implementation, cell SoC was estimated using calibrated Coulomb counting based on the measured per-cell discharge currents and the experimentally measured cell capacities listed in Table 5. This approach was selected because the proposed

controller requires reliable relative SoC feedback during reconfiguration, while the available per-cell current measurements allow the extracted charge from each cell to be directly tracked.

Before each hardware test, all cells were individually charged to the full-charge condition and held on charge for a further 30 min, during which the charger maintained the cells at full charge. The Coulomb-counting estimators were then initialized with $SoC_i(0) = 1.0$, and the subsequent cell SoCs were calculated from the measured cell currents and characterized capacities.

The Coulomb-counted SoC of each cell was calculated as:

$$SoC_i(t) = SoC_i(0) - \frac{Q_{i,used(t)}}{Q_{i.meas}} \quad (23)$$

where $Q_{i,meas}$ is the measured capacity of cell i and $Q_{i,used(t)}$ is the extracted charge, calculated from the measured cell current as:

$$Q_{i,used} = \frac{1000}{3600}\int_0^t I_i(\tau)d\tau \quad (24)$$

where I_i is the measured discharge current in amperes and $Q_{i,used}$ is expressed in mAh.

The total extracted capacity was defined as the sum of the Coulomb-counted capacities extracted from all five cells, $Q_{ext,total} = \sum_{i=1}^{5} Q_{i,used}$, while the total initial capacity was the sum of the experimentally measured cell capacities, $Q_{initial,total} = \sum_{i=1}^{5} Q_{i,meas} = 8119\ mAh$. This cell-summed extracted capacity is distinct from the load-side ampere-hours because multiple cells contribute simultaneously to the constant-3S output. Delivered load energy was calculated independently from the measured load-terminal voltage and current as $E_{load} = \frac{1}{3600}\int V_{load}(t)I_{load(t)}dt$, expressed in Wh.

The hardware test was terminated when the minimum cell SoC reached the 10% cutoff. With $Q_{ext,total} = 7236.0\ mAh$, the experimental capacity utilization was calculated as:

$$\eta_{exp} = \frac{7236.0}{8119} \times 100 = 89.12\% \quad (25)$$

This represents 99.03% of the 90% theoretical utilization limit imposed by the 10% cutoff. The SoC trajectories reached the 10% cutoff with effectively zero final spread, confirming that the target-mean controller coordinated the discharge of all five cells under the constant-3S hardware constraint.

6.5. Experimental Results

The same standardized hardware procedure was applied to each of the three evaluated controllers. Since long-duration switch-state logging increased telemetry burden, hardware comparison was based on runtime, integrated extracted capacity, utilization, final SoC spread, measured load current, measured load voltage, and delivered load energy.

The hardware comparison results for the three evaluated control strategies are summarized in Table 7.

Table 7. Hardware comparison of evaluated control strategies under the same constant-3S discharge conditions.

Controller	Runtime (s)	Mean Load Current (A)	Mean Load Voltage (V)	Extracted Capacity (mAh)	Experimental Utilization (%)	Final SoC Spread (p.p.)	Delivered Energy (Wh)
Target-mean SoC	5091.35	1.841	8.875	7236.0	89.12	0.000	23.115
Target-mean mAh	4992.6	1.849	8.843	7088.7	87.31	2.000	22.699
Parallel-based	4643.8	1.852	9.118	6576.7	81.00	19.000	21.783

The hardware results confirm the simulation trend. The proposed target-mean SoC controller achieved the longest runtime of 5091.35 s, the highest extracted capacity of 7236.0 mAh, and the highest experimental utilization of 89.12%, while reaching the cutoff with effectively zero final SoC spread. The corresponding hardware SoC trajectories, SoC spread, load voltage, and load current for the proposed target-mean SoC controller are shown in Figure 8. The target-mean mAh controller achieved lower utilization of 87.31% and ended with 2.0 p.p. SoC spread as shown in Figure 9, indicating that balancing remaining absolute capacity did not provide the best SoC-coordinated endpoint. The parallel-based controller showed the weakest performance, terminating after 4643.8 s with 81.00% utilization and 19.0 p.p. final SoC spread as shown in Figure 10. This indicated that parallel-based operation alone did not coordinate depletion of the heterogeneous cells effectively, leaving significant usable capacity unused in some cells.

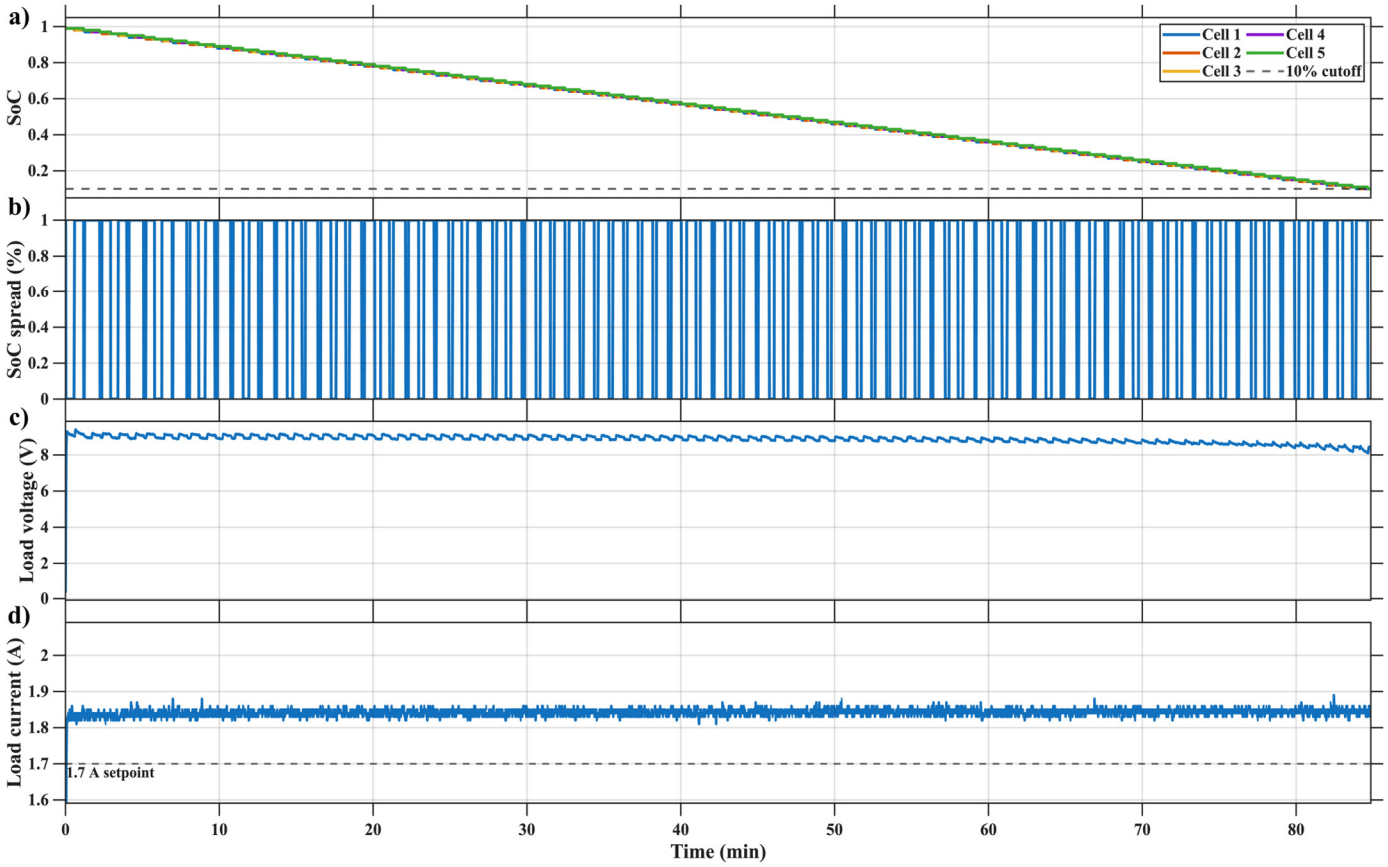

Figure 8. Hardware validation of the proposed target-mean SoC controller. (**a**) Measured cell SoC trajectories during discharge from full charge to the 10% cutoff. (**b**) Corresponding cell-to-cell SoC spread, calculated as the difference between the maximum and minimum cell SoC. (**c**) Measured load voltage, showing that the constant 3S output is maintained during controller-driven reconfiguration. (**d**) Measured load current, indicating the applied load condition throughout the discharge test.

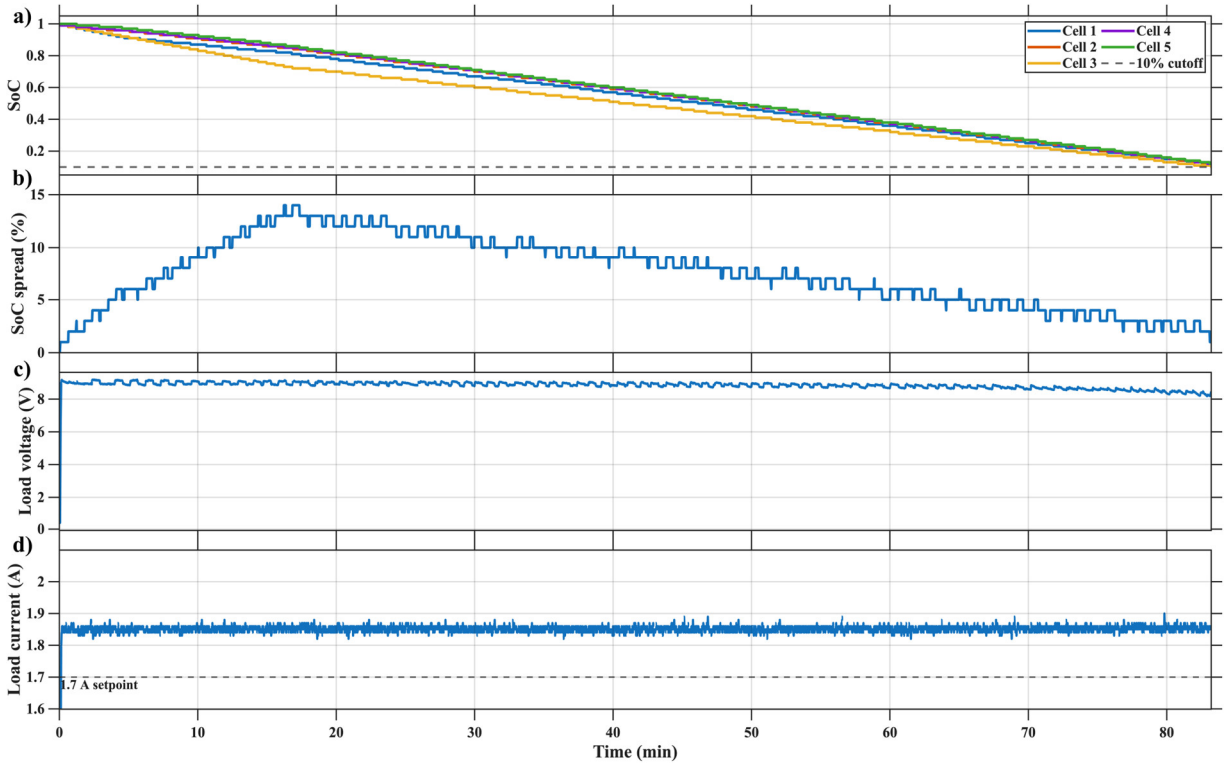

Figure 9. Hardware validation of the proposed target-mean mAh controller. (**a**) Measured cell SoC trajectories during discharge from full charge to the 10% cutoff. (**b**) Corresponding cell-to-cell SoC spread, calculated as the difference between the maximum and minimum cell SoC. (**c**) Measured load voltage, showing that the constant 3S output is maintained during controller-driven reconfiguration. (**d**) Measured load current, indicating the applied load condition throughout the discharge test.

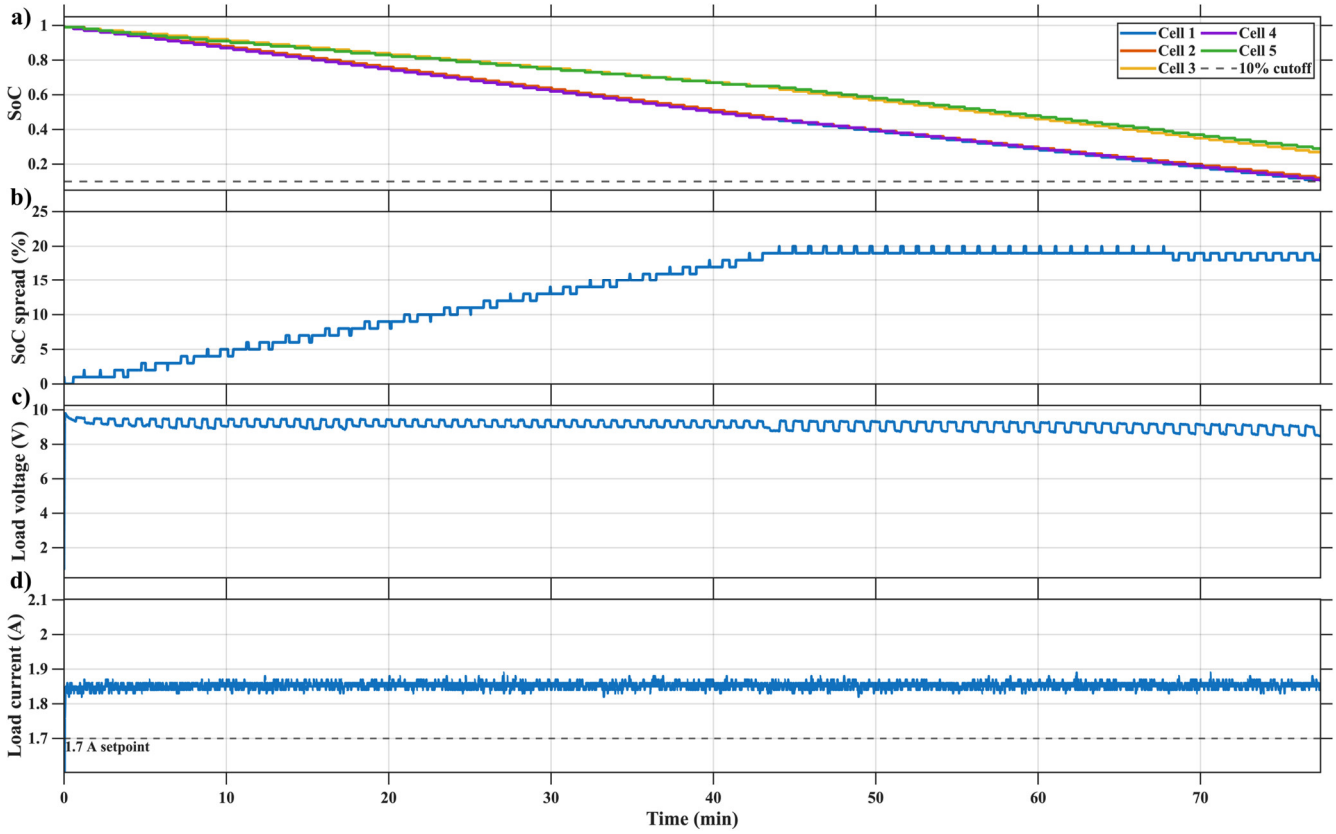

Figure 10. Hardware validation using parallel-based controller. (**a**) Measured cell SoC trajectories during discharge from full charge to the 10% cutoff. (**b**) Corresponding cell-to-cell SoC spread,

calculated as the difference between the maximum and minimum cell SoC. (**c**) Measured load voltage, showing that the constant 3S output is maintained during controller-driven reconfiguration. (**d**) Measured load current, indicating the applied load condition throughout the discharge test.

The stepped-load hardware robustness test results for the proposed target-mean SoC controller are summarized in Table 8.

Table 8. Summary of stepped-load hardware robustness test for the proposed target-mean SoC controller.

Metric	Value
Runtime (s)	5965.61
Runtime (min)	99.43
Start SoC (%)	100
End SoC range (%)	10-11
Final SoC spread (p.p.)	1.00
Final SoC standard deviation (p.p.)	0.447
Mean active load voltage (V)	9.068
Mean active load current (A)	1.557
Extracted capacity from SoC drop (mAh)	7242.2
Utilization from SoC drop (%)	89.20
Extracted capacity from current integration (mAh)	7160.4
Utilization from current integration (%)	88.19
Fraction of 90% theoretical limit (%)	97.99
Delivered load energy (Wh)	23.185

The corresponding stepped-load SoC trajectories, SoC spread, load voltage, and manually stepped-load current profile are shown in Figure 11.

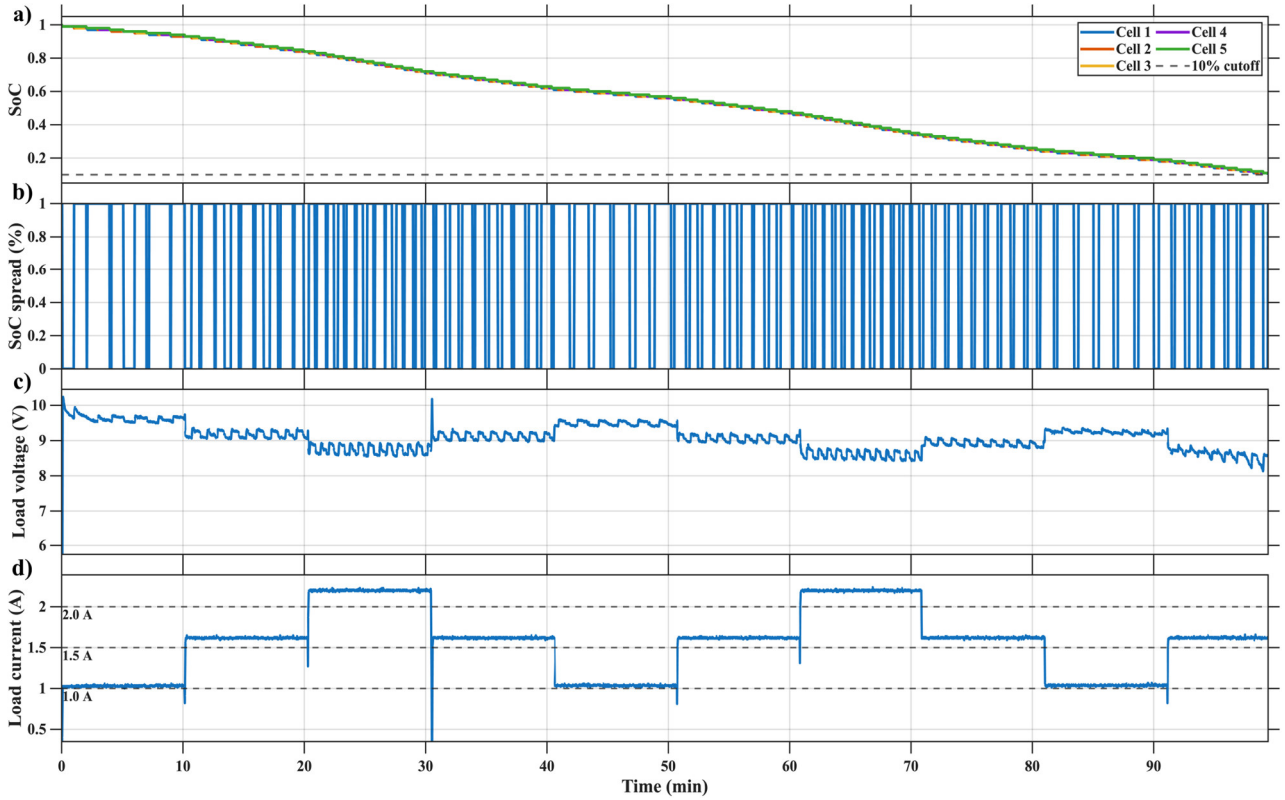

Figure 11. Hardware robustness test of the proposed target-mean SoC controller under a stepped-load-current profile. (**a**) Measured cell SoC trajectories during discharge to the 10% cutoff. (**b**)

Corresponding cell-to-cell SoC spread. (**c**) Measured load voltage, showing the maintained constant 3-S output during reconfiguration. (**d**) Measured load current under manually stepped-load profile. The current trace is smoothed for visual clarity; all capacity and energy calculations used raw measured current data.

The stepped-load hardware test confirms that the proposed controller remains effective under non-constant loading. The pack operated for 5965.61 s before reaching the 10% cutoff, with a final SoC spread of 1.00 p.p. Based on per-cell current integration, 7160.4 mAh was extracted, corresponding to 88.19% utilization, or 97.99% of the 90% theoretical limit. The delivered load energy was 23.185 Wh. These results show that the target-mean SoC strategy maintains near-maximum utilization even when the load current is varied during discharge.

7. Discussion

7.1. Why Target-Mean SoC Improves Runtime-Oriented Utilization

The results indicate that the proposed target-mean SoC controller is effective because it directly coordinates cell depletion relative to the pack-average SoC. In a reconfigurable pack operated with a per-cell cutoff constraint, useful runtime is limited by the first cell to reach the lower SoC threshold. Therefore, although equal SoC does not imply equal absolute remaining capacity in heterogeneous cells, maintaining coordinated SoC trajectories remains beneficial for runtime-oriented operation. The controller does not attempt to equalize absolute capacity; instead, it selects admissible switching actions that preferentially load cells above the pack-average SoC, thereby delaying early cutoff and improving practical utilization.

This behavior explains why the mAh-based controller did not provide the best overall result. Balancing remaining absolute capacity can drive cells toward similar residual charge values, but this does not necessarily produce the most favorable SoC endpoint when the discharge limit is imposed at 10% SoC per cell. As shown in the simulation and hardware results, the mAh-based method achieved lower utilization and larger final SoC spread than the proposed SoC-based controller. This supports the view that capacity-aware balancing is not always the most effective objective when the practical pack limit is defined by cell-level SoC thresholds.

In the full-initial-SoC capacity-mismatch case, the proposed controller achieved 90.00% utilization, equal to the theoretical maximum imposed by the 10% SoC cutoff. This is notable because prior topology-enabled hierarchical reconfiguration approaches also have targeted maximum capacity utilization [15], whereas the proposed method reaches the cutoff-defined upper limit using only real-time SoC feedback and a safety-pruned legal action set. Although direct numerical comparison is limited by differences in topology, cell number, mismatch conditions, and implementation assumptions, the result suggests that target-mean SoC control can provide a simpler runtime-oriented alternative to more complex capacity-aware reconfiguration strategies.

This is further supported by the combined mismatch case, where the target-mean SoC controller reduced remaining unused capacity from 800.00 mAh to 632.00 mAh and increased runtime by 222.20 s compared with the mAh-based controller.

7.2. Comparison with Parallel-Based Balancing

The parallel-based controller showed weaker performance in both simulation and hardware because it relies on local balancing behavior rather than global action evaluation across the full safe switching set. Under heterogeneous capacity and internal-resistance conditions, parallel interaction alone does not guarantee convergence of cell SoC trajectories. This is consistent with previous findings that capacity and internal-resistance

mismatch can cause unequal current sharing and residual imbalance in parallel-connected cells [14,16]. In the present study, the parallel-based controller retained residual imbalance in simulation, with final SoC spreads of 1.431 p.p. under internal-resistance mismatch and 3.073 p.p. under capacity mismatch, and showed the largest final SoC spread of 19.0 p.p. in hardware.

By contrast, the target-mean controller evaluates all admissible actions using the predicted effect on the full pack SoC distribution. This global action selection allows the controller to coordinate the discharge of all cells more effectively while maintaining the required constant-bus constraint. The hardware comparison further confirms this trend, where the proposed controller achieved 89.12% utilization and zero final SoC spread, compared with 87.31% and 2.0 p.p. for the mAh-based controller and 81.00% and 19.0 p.p. for the parallel-based controller.

7.3. Practical Implementation and Scalability

The proposed method is computationally lightweight because the online controller evaluates only the safety-pruned legal action set rather than the full 19-switch combination space. In the evaluated five-cell prototype, the full switching space is reduced to 132 admissible actions, all satisfying the constant-3S participation condition. These actions correspond to 10 unique three-cell participation masks, with multiple switch configurations available for some active-cell sets because of routing, bypass, and parallel-connection possibilities. This distinction is important because the controller preserves hardware-feasible switching options while using the compact *SeriesMask* representation for SoC-based decision making.

Although the present implementation is demonstrated on a five-cell constant-3S prototype, the control principle is not inherently limited to this configuration. For larger packs, the same approach can be applied by generating a safety-pruned action library for the required bus-voltage constraint and evaluating candidate actions using the corresponding cell-participation masks. In larger systems, the legal action set may grow substantially, so hierarchical grouping, modular pack segmentation, or pre-filtering of candidate actions may be required to maintain real-time operation. However, the present results show that constrained action-space pruning combined with target-mean SoC evaluation provides a practical route toward scalable reconfigurable battery control.

Muhammad et al. [6] also identified reconfiguration granularity, hardware overhead, and distributed control as open challenges for larger reconfigurable battery systems. This present work addresses these issues at prototype scale by combining safety-pruned action-space construction with a lightweight online target-mean SoC evaluation.

7.4. Limitations and Future Work

This study has several limitations. First, the hardware prototype used new LFP cells with deliberately characterized capacity and resistance variation to emulate heterogeneous pack behavior rather than cells aged through controlled second-life cycling. Future work should validate the method on genuinely aged second-life cells with wider variation in capacity, resistance, and degradation history. Second, switching losses and thermal effects were not directly measured. Although the proposed controller reduced switching activity in simulation and maintained reliable hardware operation, future work should quantify MOSFET conduction losses, switching losses, thermal behavior, and long-term reliability under repeated reconfiguration.

Third, the hardware implementation used a host PC for supervisory control while the embedded platform handled measurement acquisition, SoC estimation, and switch activation. A fully embedded implementation on the microcontroller would further strengthen practical deployability. The controller primarily depends on relative SoC

differences. Therefore, a common SoC offset does not alter the cell ranking or target-mean action selection under normal unsaturated operation. In contrast, unequal cell-specific errors caused by current-sensor bias, capacity uncertainty, or accumulated Coulomb-counting drift may influence the selected cell combination. Full-charge initialization before each test limits the initial offset, while the stable voltage behavior in Figure 8 provides qualitative support that substantial relative SoC divergence did not occur. Finally, the present work focuses on discharge operation under constant-bus constraints. Future work should extend the framework to charging, bidirectional operation, fault handling, larger pack voltages, and integration with higher-level energy-management systems.

Future work should also investigate how highly utilized reconfigurable battery systems could contribute to wider renewable-energy and grid-integration contexts. At the system level, improved storage utilization may reduce oversizing requirements and influence planning trade-offs between renewable expansion, storage deployment, and land-use constraints, which have recently been examined in clean-energy expansion models [21]. In addition, reinforcement learning and sparse-update methods could be explored for larger reconfigurable battery systems, where exhaustive evaluation of all legal switching states may become computationally expensive as the numbers of cells and switches increase [22]. Such methods may provide scalable decision-making strategies for large action spaces while retaining the safety-pruned structure developed in this work.

Future work will also investigate bidirectional operation by combining the proposed discharge-side target-mean controller with PV-side reconfigurable battery charging control. Recent work has shown that an RBS can be used for converter-less PV energy harvesting by varying the active series-cell count to track the PV maximum power point [23]. Extending the present work in this direction could enable second-life RBS modules that coordinate charging, discharging, SoC balancing, and fixed-bus formation for small DC microgrid or off-grid energy-storage applications.

8. Conclusions

This paper proposed a target-mean SoC controller for heterogeneous reconfigurable battery packs operating under constant-bus constraints. The controller uses only real-time cell SoC measurements and a safety-pruned legal action set to select switching configurations that reduce deviation from the pack-average SoC while preferentially loading cells above the mean. The online action-selection objective does not use capacity or SoH as decision variables and requires no offline optimization, although measured cell capacities are used for calibrated Coulomb-counting SoC estimation.

Simulation results showed that the proposed controller reached the cutoff-defined maximum 90% theoretical utilization limit in full-initial-SoC cases and provided the best overall compromise between runtime, utilization, residual SoC spread, and switching activity. Under individual mismatch conditions, the target-mean SoC controller achieved 90.00% utilization and zero final SoC spread, while the mAh-based and parallel-based comparison methods showed lower utilization or larger residual imbalance. In the combined mismatch case, the proposed method also achieved the highest utilization while requiring the lowest switching activity.

Hardware validation on a five-cell prototype confirmed the simulation trend. The proposed controller achieved 5091.35 s runtime, 7236.0 mAh extracted capacity, 89.12% experimental utilization, zero final SoC spread, and 23.115 Wh delivered energy. This outperformed the mAh-based and parallel-based comparison controllers under the same constant-bus discharge conditions. A stepped-load hardware test further demonstrated robustness under variable loading, achieving 88.19% utilization from current integration, equivalent to 97.99% of the cutoff-defined maximum 90% theoretical limit.

Overall, the results show that target-mean SoC-based reconfiguration can provide high utilization and runtime-oriented operation in heterogeneous battery packs without the added complexity of explicit SoH-aware balancing. The approach is therefore a promising control strategy for practical second-life reconfigurable battery systems, particularly where simple measurement requirements, safe switching constraints, and hardware feasibility are important.

Author Contributions: M.S.: Writing—original draft, Visualization, Software, Methodology, Formal analysis, Conceptualization. W.I.: Visualization, Writing—review & editing, Supervision, Funding Acquisition. M.M.: Writing—review & editing, Supervision, Resources. A.A.: Writing—review & editing, Project administration. N.R.: Writing—review & editing, Software, Investigation. A.D.N.: Writing—review & editing, Project Administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The processed simulation and hardware datasets supporting the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Nomenclature

Symbol	Description
$s_i(k)$	State of charge of cell i at time step k
$\mathbf{s}(k)$	Cell SoC vector at time step k
$\mu(k)$	Pack-average SoC at time step k
q_i	Remaining absolute charge of cell i
Q_i	Nominal or available capacity of cell i
$\mathcal{A}$	Set of admissible legal switching actions
a_j	Candidate legal switching action
N	Number of admissible legal actions
$\boldsymbol{m_j}$	Binary series participation mask for action a_j
$m_{j,i}$	Binary participation state of cell i under action a_j
$\hat{s}_{i,j}(k+1)$	Predicted next-step SoC of cell i under action a_j
$\hat{\mu}_j(k+1)$	Predicted pack-average SoC under action a_j
$J_j(k)$	Total cost of candidate action a_j
$J_{\text{err},j}$	Predicted mean-error/deviation cost term
$J_{\text{spread},j}$	Predicted SoC spread cost term
$J_{\text{sw},j}$	Switching-change cost term
$R_{mean,j}$	Reward term for selecting above-average-SoC cells
W_{err}	Weight applied to mean-error cost
W_{spread}	Weight applied to spread cost
W_{sw}	Weight applied to switching-change cost
Δ_{SOC}	One-step SoC decrement used for candidate prediction
C_{init}	Initial available pack capacity
C_{rem}	Remaining pack capacity at cutoff
P	Capacity utilization percentage
η_{exp}	Experimental capacity utilization
$I_i(t)$	Measured discharge current of cell i

$Q_{i,\text{used}}$	Extracted capacity from cell i
N_a	Number of admissible actions
N_c	Number of cells
$m_{prev,i}$	Previous participation state
$Q_{ext,total}$	Total cell-summed extracted capacity
$Q_{initial,total}$	Total measured initial capacity
E_{load}	Delivered load energy

Abbreviations

The following abbreviations are used in this manuscript:

DC	Direct current
LFP	Lithium iron phosphate
MOSFET	Metal-oxide-semiconductor field-effect transistor
RBS	Reconfigurable battery system
SoC	State of charge
SoH	State of health
EV	Electric vehicle
OCV	Open-circuit voltage
PV	Photovoltaic

References

1. Hossain, E.; Murtaugh, D.; Mody, J.; Faruque, H.M.R.; Sunny, M.S.H.; Mohammad, N. A comprehensive review on second-life batteries: Current state, manufacturing considerations, applications, impacts, barriers & potential solutions, business strategies, and policies. *IEEE Access* **2019**, *7*, 73215–73252. https://doi.org/10.1109/ACCESS.2019.2917859.
2. Braco, E.; Martín, I.S.; Sanchis, P.; Ursúa, A. Fast capacity and internal resistance estimation method for second-life batteries from electric vehicles. *Appl. Energy* **2023**, *329*, 120235. https://doi.org/10.1016/j.apenergy.2022.120235.
3. Barai, A.; Uddin, K.; Dubarry, M.; Somerville, L.; McGordon, A.; Jennings, P.; Bloom, I. A comparison of methodologies for the non-invasive characterisation of commercial Li-ion cells. *Prog. Energy Combust. Sci.* **2019**, *72*, 1–31. https://doi.org/10.1016/j.pecs.2019.01.001.
4. He, L.; Yang, Z.; Gu, Y.; Liu, C.; He, T.; Shin, K.G. SoH-Aware Reconfiguration in Battery Packs. *IEEE Trans. Smart Grid* **2018**, *9*, 3727–3735. https://doi.org/10.1109/TSG.2016.2639445.
5. Ci, S.; Lin, N.; Wu, D. Reconfigurable Battery Techniques and Systems: A Survey. *IEEE Access* **2016**, *4*, 1175–1189. https://doi.org/10.1109/ACCESS.2016.2545338.
6. Muhammad, S.; Rafique, M.U.; Li, S.; Shao, Z.; Wang, Q.; Liu, X. Reconfigurable battery systems: A survey on hardware architecture and research challenges. *ACM Trans. Des. Autom. Electron. Syst.* **2019**, *24*, 19. https://doi.org/10.1145/3301301.
7. Han, W.; Wik, T.; Kersten, A.; Dong, G.; Zou, C. Next-Generation Battery Management Systems: Dynamic Reconfiguration. *IEEE Ind. Electron. Mag.* **2020**, *14*, 20–31. https://doi.org/10.1109/MIE.2020.3002486.
8. Viswanathan, V.; Palaniswamy, L.N.; Leelavinodhan, P.B. Optimization techniques of battery packs using re-configurability: A review. *J. Energy Storage* **2019**, *23*, 404–415. https://doi.org/10.1016/j.est.2019.03.002.
9. He, L.; Kong, L.; Gu, Y.; Liu, C.; He, T.; Shin, K.G. Extending Battery System Operation via Adaptive Reconfiguration. *ACM Trans. Sens. Netw.* **2019**, *15*, 11. https://doi.org/10.1145/3284556.
10. Liu, X.; Chang, G.; Tian, J.; Wei, Z.; Zhang, X.; Wang, P. Flexible path planning-based reconfiguration strategy for maximum capacity utilization of battery pack. *J. Energy Chem.* **2023**, *86*, 362–372. https://doi.org/10.1016/j.jechem.2023.07.040.
11. Ji, F.; Liao, L.; Wu, T.; Chang, C.; Wang, M. Self-reconfiguration batteries with stable voltage during the full cycle without the DC-DC converter. *J. Energy Storage* **2020**, *28*, 101213. https://doi.org/10.1016/j.est.2020.101213.
12. Li, Y.; Yin, P.; Chen, J. Active Equalization of Lithium-Ion Battery Based on Reconfigurable Topology. *Appl. Sci.* **2023**, *13*, 1154. https://doi.org/10.3390/app13021154.
13. Yu, L.; Zhang, Y.; Huang, N.; Zhang, F. An Active Equalization Method of Battery Pack Based on Event-Triggered Consensus Algorithm. *Electronics* **2024**, *13*, 151. https://doi.org/10.3390/electronics13010151.
14. Wei, Z.; Cui, H.; Liu, X.; Li, Y.; Wang, R. Real-Time Reconfiguration-Based All-Cell Flexibility and Capacity Maximum Utilization of Second-Life Batteries. *IEEE Trans. Transp. Electrif.* **2025**, *11*, 1035–1047. https://doi.org/10.1109/TTE.2024.3399218.

15. Cui, H.; Wei, Z.; He, H.; Li, J. Novel Reconfigurable Topology-Enabled Hierarchical Equalization of Lithium-Ion Battery for Maximum Capacity Utilization. *IEEE Trans. Ind. Electron.* **2023**, *70*, 396–406. https://doi.org/10.1109/TIE.2022.3152005.
16. Gogoana, R.; Pinson, M.B.; Bazant, M.Z.; Sarma, S.E. Internal resistance matching for parallel-connected lithium-ion cells and impacts on battery pack cycle life. *J. Power Sources* **2014**, *252*, 8–13. https://doi.org/10.1016/j.jpowsour.2013.11.101.
17. Sztuka, M.; Richardson, N.; Al-Naemi, F.; Ali, A.; Ikpehai, A.; Musameh, M.; Akmal, M.; Issa, W. State-of-Charge Balancing of a Safety-Pruned Reconfigurable Battery Pack with a Constant 3S Bus Using Double Deep Q-Networks. In Proceedings of the ECCE Europe 2026 Conference, Valencia, Spain, 14–18 September 2026.
18. Chen, S.Z.; Wang, Y.; Zhang, G.; Chang, L.; Zhang, Y. Sneak Circuit Theory Based Approach to Avoiding Short-Circuit Paths in Reconfigurable Battery Systems. *IEEE Trans. Ind. Electron.* **2021**, *68*, 12353–12363. https://doi.org/10.1109/TIE.2020.3044805.
19. Garg, A.; Mou, J.; Su, S.; Gao, L. Reconfigurable battery systems: Challenges and safety solutions using intelligent system framework based on digital twins. *IET Collab. Intell. Manuf.* **2022**, *4*, 232–248. https://doi.org/10.1049/cim2.12059.
20. Wu, C.; Fu, R.; Xu, Z.; Chen, Y. Improved State of Charge Estimation for High Power Lithium-Ion Batteries Considering Current Dependence of Internal Resistance. *Energies* **2017**, *10*, 1486. https://doi.org/10.3390/en10101486.
21. Koutsandreas, D.; Ardehali, A.; Giannelos, S.; Pudjianto, D. Quantifying the Trade-Offs Between Clean-Energy Expansion and Land Requirements: Evidence from Greece. *Energies* **2026**, *19*, 2261. https://doi.org/10.3390/en19102261.
22. Kaloev, M.; Krastev, G. Comprehensive Review of Benefits from the Use of Sparse Updates Techniques in Reinforcement Learning: Experimental Simulations in Complex Action Space Environments. In Proceedings of the 2023 4th International Conference on Communications, Information, Electronic and Energy Systems (CIEES), Plovdiv, Bulgaria, 23–25 November 2023; pp. 1–7. https://doi.org/10.1109/CIEES58940.2023.10378830.
23. Richardson, N.; Sztuka, M.; Al-Naemi, F.; Ali, A.; Ikpehai, A.; Musameh, M.; Akmal, M.; Issa, W. Converter-Less PV Energy Harvesting Using a Reconfigurable Battery System Under Partial Shading. In Proceedings of the ECCE Europe 2026 Conference, Valencia, Spain, 14–18 September 2026.