Controlling Algorithm of Reconfigurable Battery for State of Charge Balancing using Amortized Q-Learning

Dominic Karnehm , Wolfgang Bliemetsrieder , Sebastian Pohlmann * , Antje Neve *

Posted Date: 2 February 2024

doi: 10.20944/preprints202402.0121.v1

Keywords: Reconfigurable Battery; Neuronal Network, SoC Balancing, Reinforcemnt Learning; Amortized Q-Learning

 doi:10.20944/preprints202402.0121.v1

Article

Controlling Algorithm of Reconfigurable Battery for State of Charge Balancing using Amortized Q-Learning

Dominic Karnehm [1], Wolfgang Bliemetsrieder [2], Sebastian Pohlmann [1,*] and Antje Neve [1,*]

[1] Electrical Engineering and Technical Informatics Department, University of the Bundeswehr Munich, Neubiberg, 85577, Germany; dominic.karnehm@unibw.de

[2] Electrical Engineering Department, University of the Bundeswehr Munich, Neubiberg, 85577, Germany; wolfgang.bliemetsrieder@unibw.de

* Correspondence: sebastian.pohlmann@unibw.de (S.P.); antje.neve@unibw.de (A.N.)

Abstract: Towards smart batteries for electric vehicles (EVs) smart algorithms to control battery packs, mainly reconfigurable batteries, have to be developed. This work proposes a reinforcement learning (RL) algorithm to balance the State of Charge (SoC) of reconfigurable batteries based on the topologies half-bridge and battery modular multilevel management (BM3). As RL algorithm, Amortized Q-learning (AQL) is implemented, which allows enourmous numbers of possible configurations of the reconfigurable battery to be controlled, as well as the combination of classical controlling approach and machine learning methods. This enables safety mechanisms in control. As a neural network of the AQL a Feedforward Neuronal Network (FNN) is implemented consisting of three hidden layers. The experimental evaluation using a 12-cell hybrid cascaded multilevel converter illustrates the applicability of the method to balance the SoC and maintain the balanced state during discharge. The evaluation shows a 20.3% slower balancing compared to the classical. Nevertheless, AQL shows great potential to be applied for multiobjective optimizations as an applicable RL algorithm for control in power electronics.

Keywords: reconfigurable battery; neuronal network; SoC balancing; reinforcemnt learning; amortized Q-learning

1. Introduction

Balance of State of Charge (SoC) in battery packs is one of the main challenges in the field of Electrical Vehicles (EVs). Therefore, many types of balancing methods have been proposed. It has to be discerned between active and passive as the two balancing methods. The passive method is a power-loose method. For this purpose, battery cells with higher SoC transfer energy through a shunt resistor to discharge [1]. This method is a cost-effective method in implementation. On the other hand, it reduces the efficiency of the battery pack. Active equalization circuits can be used in battery packs to transfer energy from cells with higher charge to those with lower charge [2–4]. Van [5] proposes an SoC balancing algorithm for in series connected battery cells. Therefore, a modified bidirectional cuk converter is used to transfer energy from cells with a higher SoC, to cells with a lower one. The usage of this algorithm has been proven during discharge and relaxation of the battery pack.

In the literature, it has been discussed to balance the voltage of the capacitor and the thermal stress [6] of a modular multilevel converter (MMC) or to optimize the efficiency of the DC-DC converter of the dual active bridge (DAB) [7,8] using Reinforcement Learning (RL) algorithms. Furthermore, the potential of different algorithms in the field of RL have been introduced for the controlling and optimization of reconfigurable batteries.

Reconfigurable batteries allow active balancing of different battery parameters, such as voltage [9], State of Temperature (SoT) [10], and SoC [11–13]. To address the problem of SoC balancing in reconfigurable batteries and modular multilevel inverter (MMI), various algorithms have been suggested. Several centralized algorithms have been proposed to balance SoC during battery relaxation

 doi:10.20944/preprints202402.0121.v1

[12,13] and charge and discharge of reconfigurable battery [11]. To reduce the communication required between the main controller and the modules, decentralized algorithms have been introduced [14–17]. Furthermore, machine learning-based algorithms have shown possible use in the field of control for reconfigurable batteries. Jiang [18] proposes a RL Deep Q-Network (DQN) to control a reconfigurable battery using the three-switch topology Battery Modular Multilevel Management (BM3) [19]. They used the reconfigurable battery as a direct current (DC) source. To evaluate the model, a simulation of a 10 module system is used. Therefore, nine modules are switched to serial mode and one to parallel mode. The neuronal network controlls which cell is switched into parallel model. Stevenson [20] uses a DQN to reduces the inbalance of SoC and the current of a reconfigurable battery with four battery cells. The reduction of inbalanced states could be observed, using the technology of DQN. The authors improve the potential to increase economic viability, and the battery sustainability is enhanced.

Mashayekh [21] introduces a decentralized online RL algorithm to control a SoC balanced modular reconfigurable battery converter based on the topology of the half-bridge converter. The focus of this method is to reduce communication between the controller and the modules, also to reduce the intercommunication between the modules to a minimum. Therefore, an algorithm is implemented based on game theory, where each module tries to maximize the reward of each self. Due to the design of the reward function, the reward of the whole system is also maximized. The algorithm shows high usability for a balanced system. In the case of an inbalanced initial state, the authors suggest the implementation of other algorithms to balance and use the decentralized algorithm to reduce communicational requirements during the balanced state.

Yang et al. [22] propose a online-learning DQN algorithm to balance the SoC of a reconfigurable battery with 64 cells and a predefined voltage output. Therefore, the authors used a neuronal network with one hidden layer. The authors compare the algorithm with the difference in SoC of a fixed configuration without any balancing mechanism.

To balance SoC of a reconfigurable battery generating different voltage levels, we proposed in [23] a Q-learning algorithm. The usage of the proposed algorithm is limited by the number of modules in the system. However, the Q-learning algorithm has shown potential to control reconfigurable batteries. Besides the balancing of SoC, furthermore State of Health (SoH), or thermal stress of the battery cells can be parameters for optimization. Toward a multiparameter optimization, the limitations of the Q-learning algorithm in the aspect of the possible number of controlled modules have to be faced. In this work an algorithm is proposed based on Amortized Q-learning (AQL), which addresses this problem. Also, the proposed algorithm enables the combination of classical algorithm-based balancing algorithms and a machine learning approach. This makes it possible to draw the positive aspects from both approaches. Providing the safety of a classical approach and the flexibility and adaptation of machine learning. Furthermore, the proposed algorithm can be applied to AC and DC reconfigurable batteries. It can be concluded that, to our knowledge, this work introduces the first reinforcement learning algorithm that allows a combination with classical algorithms to control reconfigurable batteries with variable voltage levels.

2. Reconfigurable Battery

In this paper, two topologies of MMC for a reconfigurable battery are discussed. The half-bridge [9] and BM3 [19] converter. In Figure 1 the circuits of one of each of these two converters are shown. For a converter system, these modules are interconnected to each other. A half-bridge converter module includes a battery and two MOSFET switches S_1, and S_2 and can be switched to serial and bypass modes. A BM3 module includes a battery and three MOSFETs S_1, S_2, and S_3 and can also be switched into serial and bypass modes. Additionally, it can also be switched into parallel mode.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 February 2024 doi:10.20944/preprints202402.0121.v1

(a) **(b)**

Figure 1. Electrical circuit of a half-bridge converter module (a), and a BM3 converter module (b).

As shown in Table 1 the modules can be individually switched into three modes: bypass, serial, and parallel. The two states of the switches are characterized by a fixed resistance in the on-state and an infinite resistance in the off-state. The on-state is considered with a conduction resistance of 0.55 mΩ is considered. This is based on the R_{on} resistance of the switches installed at the device under test (DUT) during experimental evaluation.

Table 1. Switch states of MOSFETs for BM3, and Half-Bridge, modeled as electrical resistors.

	S_1	S_2	S_3
Half-Bridge			
Bypass	on	off	-
Series	off	on	-
BM3			
Bypass	on	off	off
Series	off	on	off
Parallel	on	off	on

This work compares different algorithms to balance the SoC of battery cells in the proposed reconfigurable battery topologies. In this case, the SoC is defined by Coulomb counting, where the SOC at the time step $t + \Delta t$ is defined as:

$$SOC(t + \Delta t) = SOC(t) - \int_0^{\Delta t} \frac{i_B}{Q_0} dt, \tag{1}$$

where i_B is the currency of the battery, Q_0 is the capacity of the cell, and $SOC(0)$ is given as the initial SoC of the battery cell. Van [5] explains that the balance of SoC is achieved by equalizing the SoC of each cell so that the disparity between the average SoC of each cell and the general average SoC is minimized. It is defined as follows:

$$\begin{aligned} &\min \sum_{k=0}^{N} \left(SOC^k - M_{SOC} \right)^2 \\ &M_{SOC} := \frac{1}{N} \sum_{k=0}^{N} SOC^k \end{aligned} \tag{2}$$

3. RL Approach

Reinforcement Learning (RL) is one of the three main machine learning paradigms, beside the supervised and unsupervised learning. In Reinforcement Learning (RL) the agent learns to maximize cumulative reward by interacting with an environment. Due to this, training data is not required. In contrast, an environment is neccessary during the process of training. A RL problem is defined as a

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 February 2024 doi:10.20944/preprints202402.0121.v1

Markov Decision Process (MDP) containing a set of environmental states $S = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$, possible actions to interact with the environment $A = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m\}$, and the reward function $r(\mathbf{a}, \mathbf{s})$. In Figure 2 the interaction between an agent and the environment during the training process is shown. The agent chooses an action $\mathbf{a}_t$ based on the current state $\mathbf{s}_t$. Depending on the state and action, the environment returns a reward $r_t := r(\mathbf{a}_t, \mathbf{s}_t)$, determined by the reward function, and the state $\mathbf{s}_{t+1}$ [24,25].

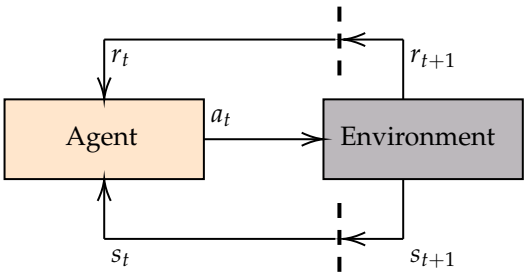

Figure 2. Interaction between Agent and Environment in Reinforcement Learning.

Generally, the actions an agent can perform are consistent across all states. A vector $\mathbf{a}$ is used to represent an element of the action space A [24]. The objective of the proposed RL algorithm in this work is to combine control methods and the possibilities provided by a RL algorithm. For the classical approach of DQN, the neuronal network takes the state of the environment as input and as output the action to take. This method limits the possibility of large discrete action spaces or the opportunity of restricting the action space. For the use case of controlling a reconfigurable battery, it is necessary to control the output voltage. Additionally, the opportunity to restrict the usage of single cells is required for safety. Van de Wiele [26] introduced the AQL a RL algorithm for enormous action spaces. This approach applies Q-learning to high-dimensional or continuous action spaces. The costly maximization of all actions $\mathbf{a}$ is replaced by the maximization of a smaller subset of possible actions sampled for a learned proposal distribution.

3.1. State Space

The state space S is defined by the normalized difference to the minimal SoC in the reconfigurable battery:

$$SOC^k_{diff} := SOC^k - \min(SOC) \tag{3}$$

$$SOC^k_{norm} := \frac{SOC^k_{diff}}{\max\left(SOC_{diff}\right)} \tag{4}$$

$$\mathbf{s} := \left[SOC^1_{norm}, SOC^2_{norm}, \ldots, SOC^N_{norm}\right], \tag{5}$$

where SOC^k is the SoC of the battery cell k, SOC^k_{diff} the difference from the minimal SoC in the system, and SOC^k_{norm} the normalized difference by the min-max scaler. Normalization allows the usage of the algorithm for the case of a large difference in SoC, as well as the case of almost complete SoC balanced battery cells.

3.2. Action Space

The action space A describes all possible actions $\mathbf{a}$:

$$A := \left\{\mathbf{a}^1, \mathbf{a}^2, \ldots, \mathbf{a}^W\right\}, \tag{6}$$

 doi:10.20944/preprints202402.0121.v1

where W is the number of all possible actions. An action is defined as the switching states m of the MMC modules of the reconfigurable battery pack.

$$\mathbf{a} := \left[m^1, \quad m^2, \quad , m^N\right] \tag{7}$$

N is the number of modules in the system and the switching state m^k of the module k is described by the following:

$$m^{(k)} = \begin{cases} 0 & \text{if Bypass} \\ 1 & \text{if Serial} \\ 2 & \text{if Parallel} \end{cases} \tag{8}$$

Furthermore, the action space A contains the subsets $A_V^{v(t)}$ to control voltage levels, where $v(t)$ is the number of battery cells switched to serial mode to generate the required voltage level at time t. Additionally, A_C can be used to restrict the acion space A. The set A_C includes excluded actions. This allows to disable a set of defined actions to enable the combination of algorithmic- and machine-learning-based control. For example, a brocken MOSFET switch or a thermally rising battery cell can cause such a restriction. Consequently, the action space A_t at time t can be defined as:

$$A_t = A_V^{v(t)} - A_C, \tag{9}$$

where $v(t)$ is the voltage level required at time t. In the case $A_t = \varnothing$, following applies:

$$v(t) := \begin{cases} v(t) - 1 & \text{if } v(t) > \frac{\text{Number of voltage levels}}{2} \\ v(t) + 1 & \text{otherwise} \end{cases} \tag{10}$$

, $v(t) - 1$ is valid to rule out a system fault. If $v(t) = 0$, $v(t) := 1$ is valid.

3.3. Reward Function

The environment provides the reward, denoted as r_t, to the agent. When the agent is in state $\mathbf{s}$ and takes action $\mathbf{a}$ at time t, the reward they receive is expressed as $r(\mathbf{s}_t, \mathbf{a}_t)$. The reward function used by the agent is defined as follows:

$$\begin{aligned} r^*(\mathbf{s}_t, \mathbf{a}_t) := &\left[\max\left(SOC_t\right) - \min\left(SOC_t\right)\right] - \\ &\left[\max\left(SOC_{t+1}\right) - \min\left(SOC_{t+1}\right)\right] \end{aligned} \tag{11}$$

$$r(\mathbf{s}_t, \mathbf{a}_t) := \alpha \cdot \max\left(0, r^*\left(\mathbf{s}_t, \mathbf{a}_t\right)\right), \tag{12}$$

where α is a bias value to increase the effect of the reward during training. This allows to ensure that the reward is not too small and does not affect the optimization of the neuronal network.

3.4. Learning Algorithm

The Q-learning function used for training input to update the parameters θ^Q of the neuronal network is defined as follows:

 doi:10.20944/preprints202402.0121.v1

$$Q_{new}(\mathbf{s}_t, \mathbf{a}_t) := (1 - \alpha) \cdot Q(\mathbf{s}_t, \mathbf{a}_t) + \alpha \cdot r_t, \tag{13}$$

where α is the learning rate of the Q-learning approach $\alpha \in [0; 1]$.

3.5. Neuronal Network and Training

The input of the network includes the SoC states SOC_{norm} of each cell, as defined in (5), and the action space A_t of the reconfigurable battery, as seen in (9). All actions of A_t are used as input to the neuronal network as a batch. The action $\mathbf{a}$ with the highest Q-value is identified by the argmax function. The size of the batch is given by B, and N describes the number of cells. The architecture of the proposed neuronal network is shown in Figure 3.

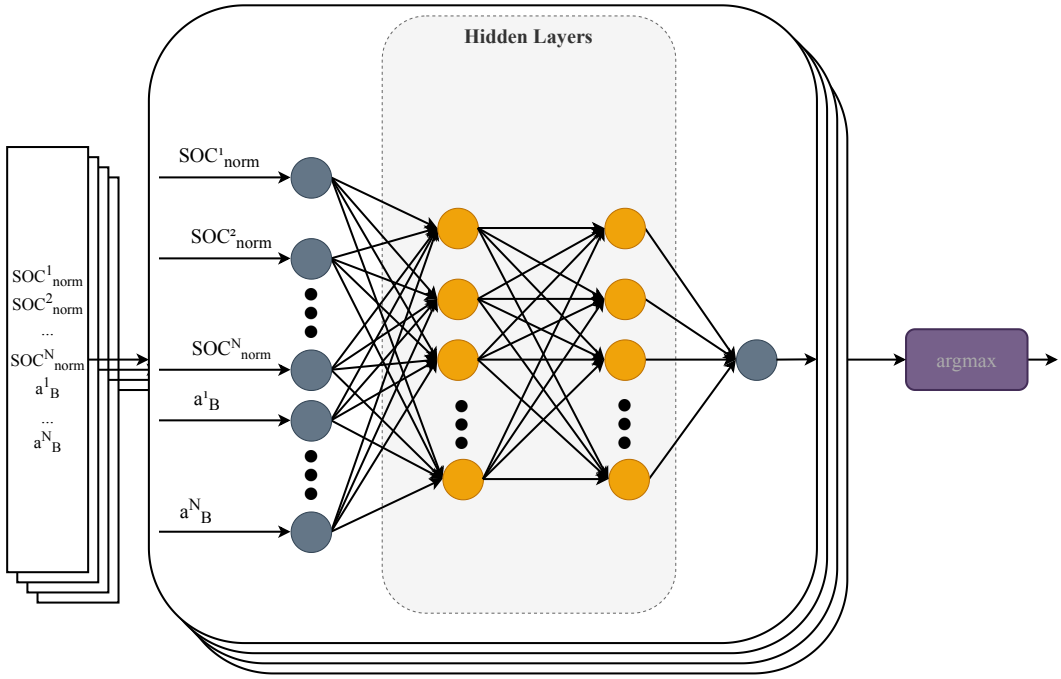

Figure 3. Architecture of neuronal network.

The proposed AQL training algorithm is described in Algorithm 1.

Algorithm 1 Amortized Q-learning (AQL) training

procedure AQL TRAINING(Network parameters θ^Q, exploration propability ϵ, epochs E, epoch length T)
  Initialize local replay buffer R
  for $e \rightarrow 1 \cdots E$ **do**
    for $t \rightarrow 1 \cdots T$ **do**
      Observe state $\mathbf{s}_t$
      if random() $\leq \epsilon$ **then**
        $a_t := a_k^t \in A_t$ ▷ Select a_t at random
      else
        $\mathbf{a}_t := \arg\max_{\mathbf{a} \in A_t} Q(\mathbf{s}_t, \mathbf{a}; \theta^Q)$
      end if
      $r_t, \mathbf{s}_{t+1} \leftarrow$ environmentStep($\mathbf{s}_t, \mathbf{a}_t$)
      $R_t \leftarrow (\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$
    end for
    Train θ^Q based on R by (13)
    Reset *environment* to random state
  end for
end procedure

 doi:10.20944/preprints202402.0121.v1

4. Description of Environment and Modle

Following the description of the training and implementation of the neuronal network, a simulated and experimental evaluation of the proposed method is provided. For training and validation, a converter system with 12 modules is assumed.

4.1. Training Environment

As an environment for training neuronal networks, the simulation of the BM3 converter in Python developed using the numba framework as just-in-time (JIT) compiler is implemented as a training environment. This framework allows simulation of a reconfigurable battery with the BM3 topology, and the half-bridge topology, as seen in Figure 1. The environment has been introduced in [27].

4.2. Model Implementation and Training

In this work, a Feedforward Neural Network (FNN) with three hidden layers is implemented and a rectified linear unit (ReLU) as the activation function between layers. The architecture of the network is shown in Table 2. The neuronal network is implemented in PyTorch 2.0.1. To implement the network, Python version 3.8 is used.

Table 2. Architecture of the model.

Layers	Model
Input Layer	Dense(24)
Hidden Layer 1	Dense(128)
	ReLU
	Dropout(0.1)
Hidden Layer 2	Dense(64)
	ReLU
	Dropout(0.1)
Hidden Layer 3	Dense(32)
	ReLU
	Dropout(0.1)
Output Layer	Dense(1)

In Figure 4 the reward for the episodes during training is illustrated. For evaluation, during training, nine random initial states of the converter with random initial SoC of each battery cell were generated. To ensure the reproducibility of the evaluation, a random seed is used. The evaluation reward is the sum of the reward, as defined in (12), of 0.1 s simulation time with a step size of $\Delta t = 10^{-5}$s. The reward of training is shown for two different topologies, half-bridge and BM3. Due to the different sizes of action spaces, the training of both models does not have to take the same amount of epochs. Therefore, 3,000 epochs and 10,000 epochs took place for the half-bridge and BM3 topology.

 doi:10.20944/preprints202402.0121.v1

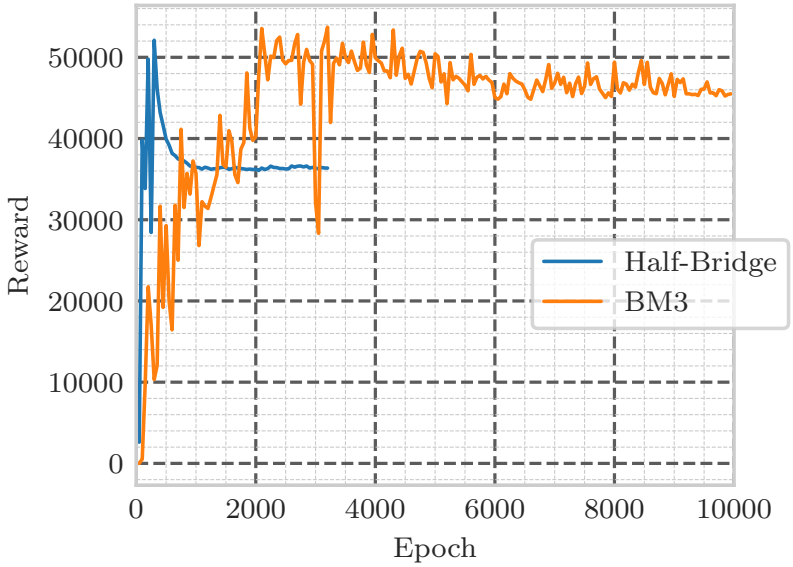

Figure 4. Reward over the training of the model for Half-Bridge (blue) and BM3 (orange) controlling.

5. Experimental Analysis and Discussion

To discuss the usability of the proposed algorithm, different scenarios have to be analyzed and discussed:

- Simulative balancing of a 12 cell BM3 converter system.
- Experimental evaluation of results with a 12 cell half-brige converter system and comparison with the balancing algorithm proposed by Zheng [28].

5.1. Simulative Evaluation

To analyze the usability of the proposed algorithm in the field of BM3 converters, a simulation is utilized. Figure 5 shows the switching, SoC, voltage, and current by discharge over 10 s at a step size of $\Delta t = 10^{-4}s$.

 doi:10.20944/preprints202402.0121.v1

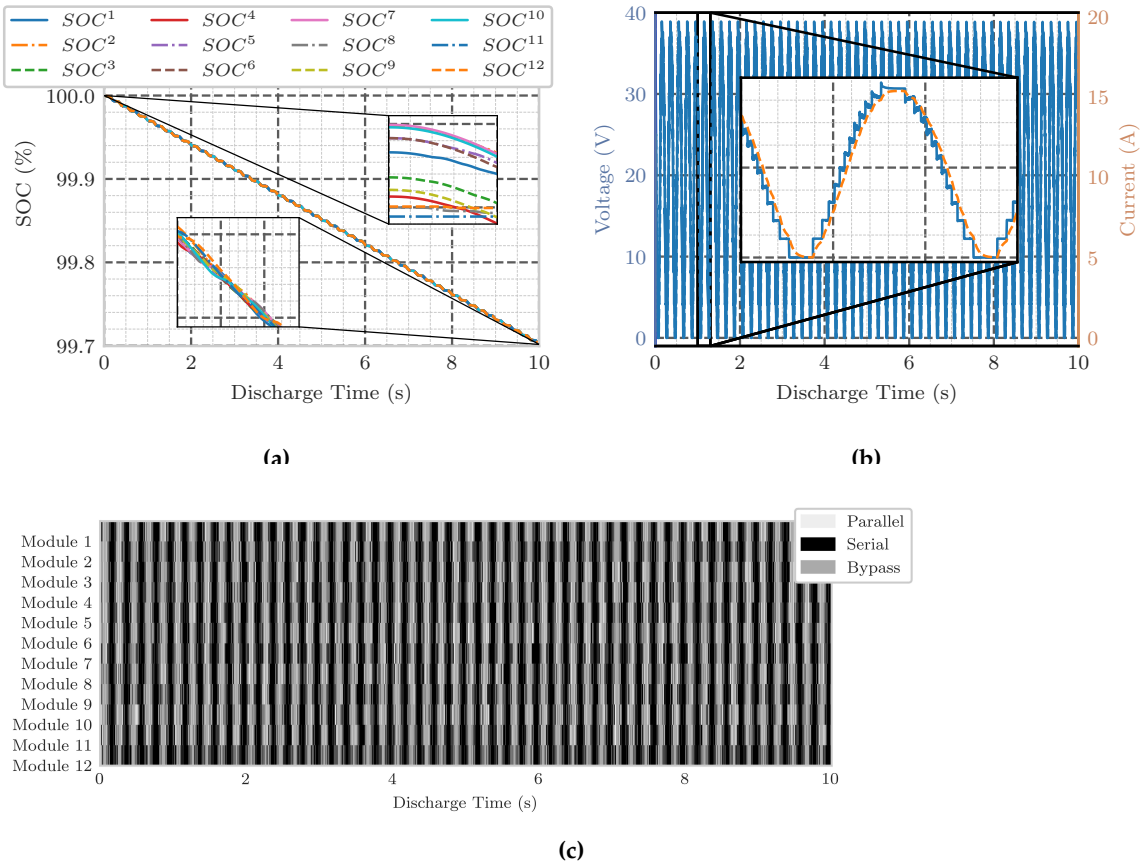

Figure 5. Simulated SoC (a), voltage and current (b), and states (c) of a BM3 converter over time using proposed AQL algorithm for 10 s and a step size for $\Delta t = 10^{-4}s$.

Firstly, the process of balancing the SoC can be observed in Figure 5a. It shows a discharge of approximately 0.3 % SoC of 12 individual battery cells, interconnected by BM3 modules. The zoom of the figure highlights the clear SoC balancing of the cells. The voltage and current corresponding to discharge can be seen in Figure 5b. The stepwise generation of voltage caused by the multilevel inverter can be detected. The algorithm utilizes serial, bypass, and parallel switch states. Analysis reveals that a battery cell switches to parallel mode in 10% of the time steps. These can be seen in Figure 5c. To generate the required voltage as a sinus wave, the cells are switched in 50% of the time steps to serial, and in conclusion in 40% the cells are switched to bypass. However, in 65% of all time steps, maximum and minimum voltages excluded, at least one battery cell is switched to parallel mode.

5.2. Experimental Evaluation

In addition to the evaluation based on the simulation of the proposed AQL algorithm to balance a BM3 converter, the algorithm is also evaluated in an experimental setting. Therefore, the setup is used as seen in Figure 6 and described as following:

- DUT: 12-cell hybrid cascaded multilevel converter [28] as reconfigurable battery module
- Raspberry Pi 4 as Controll Unit
- Lenovo ThinkPad-P15-Gen-1 as Computing Unit
- 2× Load Resistor: MAL-200 MEG 10 Ω in series
- 12× Battery Cell Simulator: Rohde & Schwarz NGM202 Power Supply

 doi:10.20944/preprints202402.0121.v1

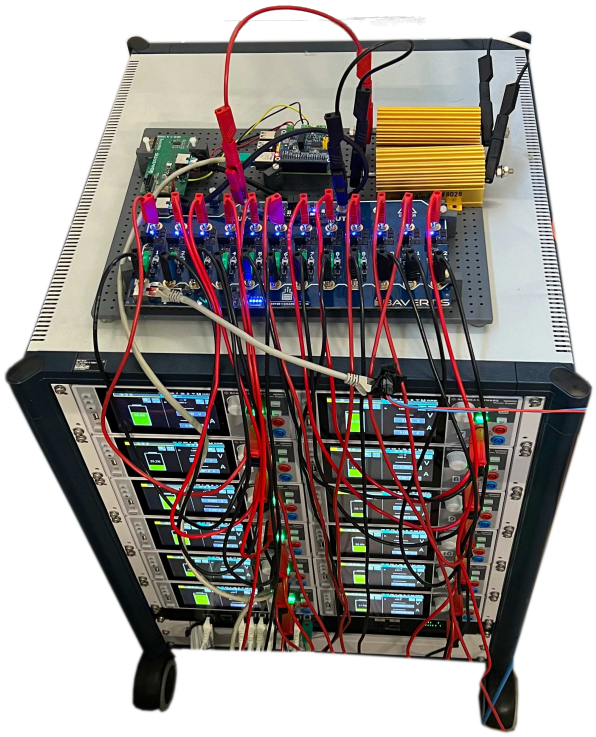

Figure 6. Experimental setup with a 12-cell hybrid cascaded multilevel converter as DUT.

To ensure a reproducible setup for the evaluation, a power supply is used for battery simulation. Furthermore, this enables the establishment of a seed randomly chosen initial SoC between 100% and 70% of the cells. As seed the experimental ID is set. During the experiment, cells are discharged from the initial SoC to 20%. The access of the battery cell data, the SoC of each cell, and the neuronal network is executed in the computing unit. As a control unit, a Raspberry Pi 4 is used, connected over the Controller Area Network (CAN) bus to the device under test (DUT). The control unit, the computing unit, and the battery cell simulators are connected over Ethernet for communication. The AQL algorithm is executed in the computing unit. The circuit board switching states are set by the control unit on the basis of the actions determined.

In Figure 7a the balancing process can be seen. The discharge is taken place for 100 s. It can be observed that the balanced state is reached after 71.9 s at SoC of 54 %. In this paper, the balanced state is defined as a maximum difference in SoC of 1 % in cells with a capacity of 0.1 A h. Furthermore, an approximately linear tapper of the discharge curves of the single cells can be detected.

 doi:10.20944/preprints202402.0121.v1

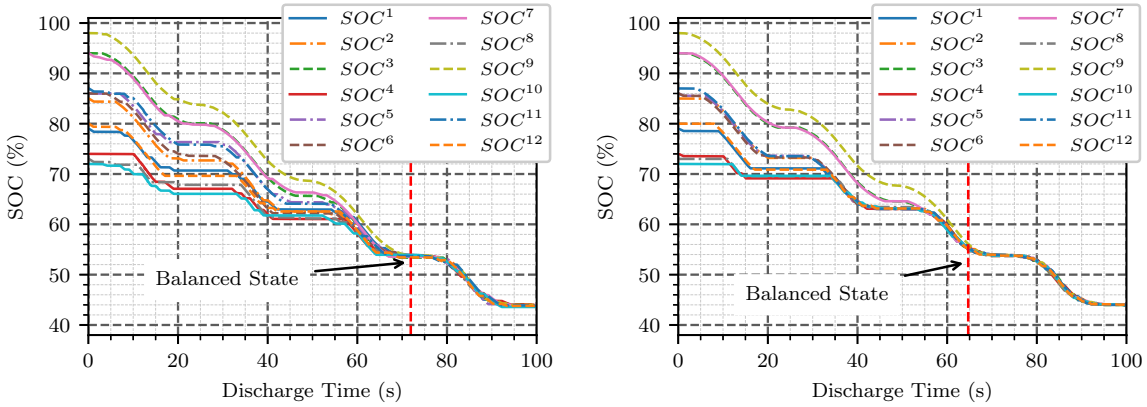

Figure 7. SoC balancing using the proposed AQL algorithm (a) and switch-max algorithm (b); Balanced state after 71.9 s and 64.7 s (red) of discharge.

To evaluate the proposed method, the balancing algorithm proposed by Zheng et al. [28] is also examined. The algorithm is defined as follows, as a cell with a higher SoC can be discharged more, those will be used more and thus the energy utilization ration is increased . Therefore, the n cells with the highest SoC are switched on, where n is the number of cells required to reach the requested voltage level. Due to this, the algorithm is following named as switching-max. The results of the balancing process can be seen in Figure 7b. The initial SoC of each cell is set identically to ensure the same conditions, as for the evaluation of the AQL algorithm. This algorithm reaches a balanced state, SoC difference below 0.1 percent, after 64.7 s at SoC of 55.4 %.

For a valid evaluation of the proposed method and a discussion of the results compared to the algorithm proposed by Zheng et al. [28] the experiment has taken place 50 times. Therefore, the experiment has taken place with 50 different random generated, to ensure comparability with the ID of the experiment as seed, initial SoC of each battery cell. The discharge time required to reach the balanced state, defined by a maximum difference of 1 % SoC between the cells, of each experiment can be seen in Figure 8. In addition, it shows the mean (solid line) and standard deviation (shaded region) of both methods. It can be observed that the mean time required to balance the switching max algorithm is 12.0 s or 20.3 % faster compared to the proposed AQL algorithm. A mean time of 70.9 s with a standard deviation of 11.9 s can be observed for the AQL method. Furthermore, the switching-max algorithm shows 58.9 s $\pm$ 10.0 s time for balancing.

 doi:10.20944/preprints202402.0121.v1

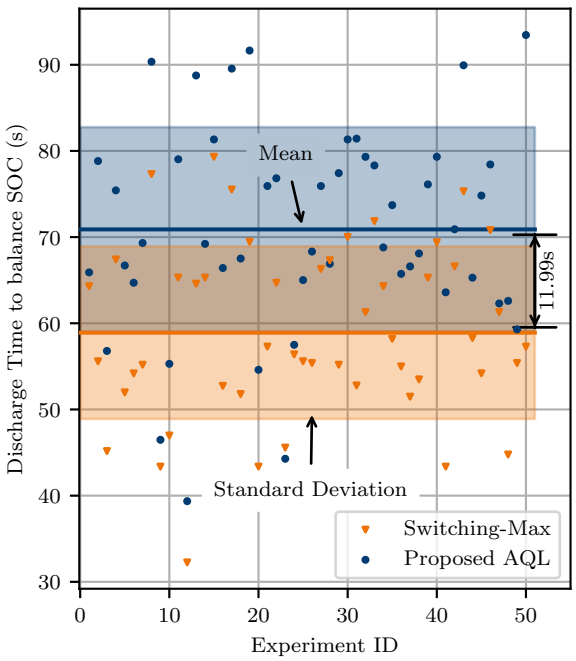

Figure 8. Comparasion of the required time to reach balanced state using the switching-max algorithm and the proposed AQL algorithm over 50 experiments.

6. Discussion

Evaluation based on the simulation of a BM3 converter and the experimental setup of the half-bridge converter shows the general usability of the proposed method. The balance process can be observed in both scenarios. Furthermore, after the balanced state is reached, a steady discharge of all cells can be seen. This suggests that the algorithm can be used for balancing, as well as for controlling during the balanced state. The computational cost is a significant component mainly for the BM3 topology. Without limiting the range of possible actions, up to 19,448 valid configurations must be examined to choose the best by the neuronal network. This is the case for the five cells in serial configuration, the voltage level with the largest amount of valid configurations. The limitation of the action space provided in Equation (9) can help reduce this problem and improve computational cost and time. Additionally, the setup of a dynamic action space can also be used to reduce the number of switchings per time step.

Comparison between the proposed AQL SoC balancing algorithm and the switching-max algorithm proposed by Zheng [28] shows a 12.0 s or 20.3 % slower balance for the AQL algorithm. This has been observed in the comparison of 50 individual experiments per algorithm, as shown in Figure 8. However, the main goal of this work is to introduce an algorithm using a machine learning method that allows the control of reconfigurable batteries with variable voltage output. The algorithm shows applicability for different types of topologies. Furthermore, the proposed algorithm enables the combination of classical algorithms with the proposed AQL algorithm by allowing dynamic action spaces as introduced in Equation (9). No comparison has been made with other RL algorithms, such as [18,20,22], because they are leaking in the ability to apply for different voltage levels. Therefore, per voltage level, a separate neuronal network would be required.

7. Conclusion and Outlook

This work introduced Amortized Q-learning (AQL), a reinforcement learning algorithm, to balance SoC for reconfigurable batteries using BM3 or half-bridge topology. It improves the Q-learning

 doi:10.20944/preprints202402.0121.v1

algorithm that uses reinforcement learning to balance SoC for these topologies [23]. The former algorithm is limited in the number of modules to be controlled. This limitation is faced in this work. Furthermore, the proposed algorithm allows the combination of algorithm- and machine learning-based control of the reconfigurable battery. Thus, individual cells are not used to switch states due to broken MOSFET-switches or because of safety reasons, such as a possible thermal runaway of cells. The effects of the combination of machine learning and the classical approach have to be discussed in the following studies. This work introduces an algorithm that uses reinforcement learning to control a reconfigurable battery with variable voltage output and the possibility of restricting possible actions selected by the algorithm, to ensure device safety.

The algorithm is evaluated in an experimental setup of a hybrid cascaded multilevel converter and as a simulation of a BM3 converter. This suggests that the applicability of the algorithm is possible for the scenario described. The balancing process is shown in Figure 5a, and Figure 7a using the proposed algorithm. A comparison with the algorithm proposed by Zheng [28] shows a 20.8 % slower balancing. In addition, the proposed algorithm requires higher complexity and computational cost during execution. However, due to the lower switching time requirements, reconfigurable batteries as DC source can be a field of application of the proposed algorithm. Therefore, in the following work, it must be evaluated whether AQL models can optimize multidimensional problems, such as SoC balancing and loss reduction and thermal balance of battery cells. Therefore, as a neuronal network, a variant of a recurrent neural network (RNN) architecture must be analyzed. This kind of neuronal network is developed to analyze time-series data. This characteristic allows controlling with time relatedness.

Furthermore, a comparison with the reinforcement learning algorithm that limits the communication required between different components of the reconfigurable battery proposed by Mashayekh [11] can indicate the various fields of application of both algorithms and the positive and negative aspects of them.

Author Contributions: Conceptualization, D.K. and A.N.; methodology, D.K.; software, D.K.; validation, D.K. and W.B.; formal analysis, D.K. and W.B.; data curation, D.K.; writing—original draft preparation, D.K.; writing—review and editing, W.B., S.P., A.N.; visualization, D.K.; supervision, A.N.; funding acquisition, A.N.; All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by dtec.bw – Digitalization and Technology Research Center of the Bundeswehr which we gratefully acknowledge. dtec.bw is funded by the European Union – NextGenerationEU.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

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

Abbreviations

The following abbreviations are used in this manuscript:

 doi:10.20944/preprints202402.0121.v1

AQL	Amortized Q-learning
AC	Alternating current
BM3	Battery Modular Multilevel Management
BMS	Battery Management System
DC	Direct Current
DUT	Device Under Test
DQN	Deep Q-Network
EVs	Electrical Vehicles
FNN	Feedforward Neural Network
MDP	Markov Decision Process
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
MMI	Modular Multilevel Inverter
MMC	Modular Multilevel Converter
RL	Reinforcement Learning
SoC	State of Charge
SoH	State of Health
SoT	State of Temperature

References

1. Gallardo-Lozano, J.; Romero-Cadaval, E.; Milanes-Montero, M.I.; Guerrero-Martinez, M.A. A novel active battery equalization control with on-line unhealthy cell detection and cell change decision. *Journal of Power Sources* **2015**, *299*, 356–370.
2. Zhang, Z.; Zhang, L.; Hu, L.; Huang, C. Active cell balancing of lithium-ion battery pack based on average state of charge. *International Journal of Energy Research* **2020**, *44*, 2535–2548. doi:10.1002/er.4876.
3. Ghaeminezhad, N.; Ouyang, Q.; Hu, X.; Xu, G.; Wang, Z. Active Cell Equalization Topologies Analysis for Battery Packs: A Systematic Review. *IEEE Transactions on Power Electronics* **2021**, *36*, 9119–9135. doi:10.1109/TPEL.2021.3052163.
4. Cao, Y.; Abu Qahouq, J.A. Hierarchical SOC Balancing Controller for Battery Energy Storage System. *IEEE Transactions on Industrial Electronics* **2021**, *68*, 9386–9397. doi:10.1109/TIE.2020.3021608.
5. Van, C.N.; Vinh, T.N.; Ngo, M.D.; Ahn, S.J. Optimal SoC Balancing Control for Lithium-Ion Battery Cells Connected in Series. *Energies* **2021**, *14*. doi:10.3390/en14102875.
6. Jung, J.H.; Hosseini, E.; Liserre, M.; Fernández-Ramírez, L.M. Reinforcement Learning Based Modulation for Balancing Capacitor Voltage and Thermal Stress to Enhance Current Capability of MMCs. 2022 IEEE 13th International Symposium on Power Electronics for Distributed Generation Systems (PEDG), 2022, pp. 1–6. doi:10.1109/PEDG54999.2022.9923188.
7. Tang, Y.; Hu, W.; Cao, D.; Hou, N.; Li, Y.; Chen, Z.; Blaabjerg, F. Artificial Intelligence-Aided Minimum Reactive Power Control for the DAB Converter Based on Harmonic Analysis Method. *IEEE Transactions on Power Electronics* **2021**, *36*, 9704–9710. doi:10.1109/TPEL.2021.3059750.
8. Tang, Y.; Hu, W.; Xiao, J.; Chen, Z.; Huang, Q.; Chen, Z.; Blaabjerg, F. Reinforcement Learning Based Efficiency Optimization Scheme for the DAB DC–DC Converter With Triple-Phase-Shift Modulation. *IEEE Transactions on Industrial Electronics* **2021**, *68*, 7350–7361. doi:10.1109/TIE.2020.3007113.
9. Tashakor, N.; Li, Z.; Goetz, S.M. A generic scheduling algorithm for low-frequency switching in modular multilevel converters with parallel functionality. *IEEE Transactions on Power Electronics* **2020**, *36*, 2852–2863.
10. Kristjansen, M.; Kulkarni, A.; Jensen, P.G.; Teodorescu, R.; Larsen, K.G. Dual Balancing of SoC/SoT in Smart Batteries Using Reinforcement Learning in Uppaal Stratego. IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society, 2023, pp. 1–6. doi:10.1109/IECON51785.2023.10311828.
11. Mashayekh, A.; Kersten, A.; Kuder, M.; Estaller, J.; Khorasani, M.; Buberger, J.; Eckerle, R.; Weyh, T. Proactive SoC Balancing Strategy for Battery Modular Multilevel Management (BM3) Converter Systems and Reconfigurable Batteries. 2021 23rd European Conference on Power Electronics and Applications (EPE'21 ECCE Europe), 2021, pp. P.1–P.10. doi:10.23919/EPE21ECCEEurope50061.2021.9570543.
12. Huang, H.; Ghias, A.M.; Acuna, P.; Dong, Z.; Zhao, J.; Reza, M.S. A fast battery balance method for a modular-reconfigurable battery energy storage system. *Applied Energy* **2024**, *356*, 122470.

13. Han, W.; Zou, C.; Zhang, L.; Ouyang, Q.; Wik, T. Near-fastest battery balancing by cell/module reconfiguration. *IEEE Transactions on Smart Grid* **2019**, *10*, 6954–6964.
14. McGrath, B.P.; Holmes, D.G.; Kong, W.Y. A decentralized controller architecture for a cascaded H-bridge multilevel converter. *IEEE transactions on industrial electronics* **2013**, *61*, 1169–1178.
15. Xu, B.; Tu, H.; Du, Y.; Yu, H.; Liang, H.; Lukic, S. A distributed control architecture for cascaded H-bridge converter with integrated battery energy storage. *IEEE Transactions on Industry Applications* **2020**, *57*, 845–856.
16. Pinter, Z.M.; Papageorgiou, D.; Rohde, G.; Marinelli, M.; Træholt, C. Review of Control Algorithms for Reconfigurable Battery Systems with an Industrial Example. 2021 56th International Universities Power Engineering Conference (UPEC), 2021, pp. 1–6. doi:10.1109/UPEC50034.2021.9548259.
17. Morstyn, T.; Momayyezan, M.; Hredzak, B.; Agelidis, V.G. Distributed control for state-of-charge balancing between the modules of a reconfigurable battery energy storage system. *IEEE Transactions on Power Electronics* **2015**, *31*, 7986–7995.
18. Jiang, B.; Tang, J.; Liu, Y.; Boscaglia, L. Active Balancing of Reconfigurable Batteries Using Reinforcement Learning Algorithms. 2023 IEEE Transportation Electrification Conference & Expo (ITEC), 2023, pp. 1–6. doi:10.1109/ITEC55900.2023.10187076.
19. Kuder, M.; Schneider, J.; Kersten, A.; Thiringer, T.; Eckerle, R.; Weyh, T. Battery modular multilevel management (bm3) converter applied at battery cell level for electric vehicles and energy storages. PCIM Europe digital days 2020; International Exhibition and Conference for Power Electronics, Intelligent Motion, Renewable Energy and Energy Management. VDE, 2020, pp. 1–8.
20. Stevenson, A.; Tariq, M.; Sarwat, A. Reduced Operational Inhomogeneities in a Reconfigurable Parallelly-Connected Battery Pack Using DQN Reinforcement Learning Technique. 2023 IEEE Transportation Electrification Conference & Expo (ITEC), 2023, pp. 1–5. doi:10.1109/ITEC55900.2023.10187040.
21. Mashayekh, A.; Pohlmann, S.; Estaller, J.; Kuder, M.; Lesnicar, A.; Eckerle, R.; Weyh, T. Multi-Agent Reinforcement Learning-Based Decentralized Controller for Battery Modular Multilevel Inverter Systems. *Electricity* **2023**, *4*, 235–252.
22. Yang, F.; Gao, F.; Liu, B.; Ci, S. An adaptive control framework for dynamically reconfigurable battery systems based on deep reinforcement learning. *IEEE Transactions on Industrial Electronics* **2022**, *69*, 12980–12987.
23. Karnehm, D.; Pohlmann, S.; Neve, A. State-of-Charge (SoC) Balancing of Battery Modular Multilevel Management (BM3) Converter using Q-Learning. The 15th Annual IEEE Green Technologies (GreenTech) Conference, 2023.
24. Jang, B.; Kim, M.; Harerimana, G.; Kim, J.W. Q-Learning Algorithms: A Comprehensive Classification and Applications. *IEEE Access* **2019**, *7*, 133653–133667. doi:10.1109/ACCESS.2019.2941229.
25. Mirchevska, B.; Hügle, M.; Kalweit, G.; Werling, M.; Boedecker, J. Amortized Q-learning with Model-based Action Proposals for Autonomous Driving on Highways. 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 1028–1035. doi:10.1109/ICRA48506.2021.9560777.
26. Van de Wiele, T.; Warde-Farley, D.; Mnih, A.; Mnih, V. Q-Learning in enormous action spaces via amortized approximate maximization. Technical Report arXiv:2001.08116, arXiv, 2020.
27. Karnehm, D.; Sorokina, N.; Pohlmann, S.; Mashayekh, A.; Kuder, M.; Gieraths, A. A High Performance Simulation Framework for Battery Modular Multilevel Management Converter. 2022 International Conference on Smart Energy Systems and Technologies (SEST), 2022, pp. 1–6. doi:10.1109/SEST53650.2022.9898406.
28. Zheng, Z.; Wang, K.; Xu, L.; Li, Y. A Hybrid Cascaded Multilevel Converter for Battery Energy Management Applied in Electric Vehicles. *IEEE Transactions on Power Electronics* **2014**, *29*, 3537–3546. doi:10.1109/TPEL.2013.2279185.