qTSL: A Multilayer Control Framework for Managing Capacity, Temperature, Stress, and Losses in Hybrid Balancing Systems

Ricardo de Castro, *Senior Member, IEEE*, Hélder Pereira, *Graduate Student Member, IEEE*, Rui Esteves Araújo, *Member, IEEE*, Jorge Varela Barreras, *Member, IEEE*, and Herschel C. Pangborn, *Member, IEEE*

***Abstract*— This work deals with the design and validation of a control strategy for hybrid balancing systems (HBSs), an emerging concept that joins battery equalization and hybridization with supercapacitors (SCs) in the same system. To control this system, we propose a two-layer model predictive control (MPC) framework. The first layer determines the optimal state-of-charge (SoC) reference for the SCs considering long load forecasts and simple pack-level battery models. The second MPC layer tracks this reference and performs charge and temperature equalization, employing more complex module-level battery models and short load forecasts. This division of control tasks into two layers, running at different time scales and model complexities, enables us to reduce computational effort with a small loss of control performance. Experimental validation in a small-scale laboratory prototype demonstrates the effectiveness of the proposed approach in reducing charge, temperature, and stress in the battery pack.**

***Index Terms*— Battery balancing, energy management, hybrid energy storage, model predictive control (MPC).**

I. Introduction

ELECTRIFICATION is currently envisaged as a key technology to develop vehicles that are more energy efficient and lower in their environmental impact, especially when integrated with renewable energies. Electrification is also aligned with long-term societal and policy goals that aspire a full transition to electric and zero-emission transportation within the 2030–2050 timeframe [1]. However, vehicular energy storage systems pose several barriers that must be overcome to meet this long-term goal. Batteries, currently the dominant technology, struggle to offer high lifespan, large specific energy and power, and fast charge rates at an affordable cost [2].

Battery packs are also prone to capacity, thermal, and aging imbalances in their cells, which appear due to manufacturing tolerances [3], variations in operation conditions [4], and faults [5]. Cell-to-cell variations limit the energy and power delivery that can be transferred to/from the battery pack. This limitation is more severe in series connections of cells, leading to unbalanced charge/discharge of the cells' energy, accelerated degradation, and premature faults, known as the *weakest cell problem* [6]. To overcome this problem, several approaches have been proposed, ranging from *passive balancing*, *active balancing*, and *reconfigurable batteries* to *multifunctional balancing*.

In *passive balancing*, a dissipative network of resistances controlled by switches is connected in parallel with battery cells. This network achieves equalization by dissipating energy of the most charged cells. Simplicity and low cost are the most attractive features of this approach, making it the preferable choice in the industry, but at the expense of higher energy losses [7]. More efficient solutions, based on *active balancing* circuits, have also been considered in the literature. They rely on power converters to transfer energy within the battery pack, either from one cell to another [8], [9], from a cell to the battery pack [10], or from a cell to a small energy buffer, such a capacitor or inductor [9]. As a result, equalization of voltage, charge, and temperature of battery packs can be effectively performed, reducing stress in the weakest cells and increasing driving range of the vehicle [9]. Higher costs and volume represent the main drawback of these approaches. *Reconfigurable batteries* are another solution to the weakest cell problem. These approaches rearrange the structure of the battery pack through a network of switches, which, in addition to charge and temperature balancing, also enables isolation of faulty cells [11].

More recently, there has been a trend toward the development of *multifunctional balancing* electronics [12]. The key idea is to exploit the balancing hardware to perform more functions than just equalization. This integration of functions decreases the number of components, volume, and

Manuscript received November 8, 2020; revised May 19, 2021; accepted July 14, 2021. Manuscript received in final form August 5, 2021. Recommended by Associate Editor D. Raimondo. The work of Jorge Varela Barreras was supported by EPSRC Faraday Institution Multi-Scale Modelling Project EP/S003053/1 under Grant FIRG003. *(Corresponding author: Ricardo de Castro.)*

Ricardo de Castro is with the Department of Mechanical Engineering, University of California at Merced, Merced, CA 95343 USA (e-mail: rpintodecastro@ucmerced.edu).

Hélder Pereira and Rui Esteves Araújo are with INESC TEC and the Faculty of Engineering, University of Porto, 4200-465 Porto, Portugal (e-mail: up201503799@fe.up.pt; raraujo@fe.up.pt).

Jorge Varela Barreras is with the Department of Mechanical Engineering, Imperial College London, London SW7 1AY, U.K., and also with the Faraday Institution, Didcot OX11 0RA, U.K. (e-mail: jvarelab@ic.ac.uk).

Herschel C. Pangborn is with the Department of Mechanical Engineering, The Pennsylvania State University, University Park, PA 16801 USA (e-mail: hcpangborn@psu.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCST.2021.3103483.

Digital Object Identifier 10.1109/TCST.2021.3103483

costs. For example, Wang and Preindl [13] investigated a power conversion architecture that simultaneously performs battery equalization and charging of a low-voltage auxiliary battery; this approach eliminates the need for additional dc/dc converters to supply the vehicle's low-voltage system. Singer *et al.* [14] proposed a modular multilevel converter topology that can dynamically connect battery cells in series or parallel. In addition to balancing functions, this topology is able to generate output voltage with variable shape, which eliminates the need to install bulky and costly voltage converters between the battery pack and the load. Extensions of this topology to handle multiphase loads have been recently reported [15].

Another *multifunction balancing* concept based on a hybrid balancing system (HBS) was proposed in [16]. Its power electronics are able to perform battery equalization functions and hybridization with additional storage systems, such as supercapacitors (SCs) (see Fig. 1). Due to the integration of these two functions into a single system, lower costs of battery-SCs hybridization can be achieved. As a result, SCs, with high power density and durability, can be integrated into the energy storage system to support batteries during rapid power bursts and reduce their degradation [17].

Multifunctional balancing solutions, such as the HBS, present integration opportunities, but also control challenges. These systems require the coordination of multiple power converters, the enforcement of safety constraints (e.g., minimum/maximum state of charge or voltages), and the exploitation of tradeoffs between multiple technical and economic objectives. To handle all these requirements, multiobjective constrained optimization approaches are often employed. For example, de Castro *et al.* [16] developed an optimal control framework that relies on convex formulations and full preview of the driving cycle for HBS control. Despite being able to provide optimal solutions, the computational effort of this approach is high, making it infeasible for implementation in embedded control units. Receding horizon approaches, such as model predictive control (MPC), are also often used to control balancing electronics. These approaches usually employ shorter prediction horizons and reduced-order electrothermal models to balance charge [18], [19], temperature [20], [21], and/or voltage [22] of the battery pack in real time.

In this work, we investigate a receding horizon control framework for HBS. This framework seeks to minimize charge imbalances (q), temperature imbalances (T), battery stress factors (S), and energy losses (L). The first criterion (q) is important to mitigate operational constraints induced by the weakest cell, e.g., premature stopping of battery discharge due to lower voltage/charge limits reached by the weakest cell. The second criterion (T) promotes uniform temperatures and attenuates thermal gradients in the battery pack, one of the main sources for aging variability in the cells [23]. The third criterion (S) focuses on stress factors relevant to battery degradation, such as load current and high temperatures [24]. The last criterion (L) avoids energy losses and inefficiencies.

While q, T, and L goals have been previously investigated in [16], [20], and [21], the integration of stress factors (S) within MPC-based HBSs has received less attention in the literature to date. To fully optimize the operation of the HBS (especially stress), longer prediction horizons are required. This increases computational complexity and complicates real-time deployment of single-layer (SL) MPC frameworks, i.e., where all control goals (q, T, S, and L) are tackled in an SL optimization problem.

Hierarchical MPC frameworks represent a promising approach to reduce design and computational complexity. These frameworks decompose the overall control design into multiple layers, each operating with a different update rate, prediction horizon, and model complexity [25]. Information exchange between layers coordinates their decision-making. Spurred by these ideas, we propose a hierarchical MPC scheme for addressing (*q, T, S, L*) control goals in HBSs. Our concept performs a functional decomposition of the control problem into two layers (see Fig. 1). The higher layer generates a reference for the SCs' state of charge (SoC) that provide a good tradeoff between *S* and *L* criteria; it uses simple *pack-level* models discretized with long sample times. The lower layer tracks the SoC reference and equalizes charge and temperature (*q, T*); it employs *module-level* prediction models discretized with shorter sample times. The main advantage of this multilayer (ML) control framework is the reduction of total computational effort, while having a small penalization in the overall control performance, when compared to SL MPC configurations.

It is worth mentioning that ML MPC frameworks have been previously applied to the control of batteries, in particular for thermal and electrical energy management in automotive [26] and aircraft [27] domains. However, these previous works focused mainly on *pack-level* control of batteries, neglecting charge or capacity imbalances. To address this gap, our work goes one step further and investigates the potential of ML MPC for *module-level* control of batteries while coping with multiple goals (*q, T, S, L*), physical limits, and real-time computational constraints. The effectiveness of the proposed concept is experimentally demonstrated in a small-scale HBS prototype, which represents another contribution of this work. A preliminary version of this work was presented in [28]; it is extended here by expanding the analysis and tuning guidelines of the ML MPC framework and providing experimental validation of the concept in a small-scale prototype.

II. Modeling and Problem Formulation

A. Overview of the HBS

As shown in Fig. 1, the HBS is composed of a battery pack, SCs, the balancing circuits, and the load. The battery pack contains n battery modules, which can be formed by one or more battery cells. Each module is connected to a balancing circuit that transfers energy to/from other modules and SCs. We assume that inner current loops are embedded within the balancing circuits. These inner loops manipulate the converters' switches such that balancing currents deviated from the battery modules $\mathbf{i}_B = \left[i_{B,1}\ , \ldots ,\ i_{B,n}\right]^T$ follow the reference setpoint $\mathbf{u} = \left[i^*_{B,1}\ , \ \ldots ,\ \ i^*_{B,n}\right]^T$. The inclusion of inner current loops brings several advantages. For example, the design of higher control layers can rely on low-order models for the closed-loop response of the current loops, instead of complex nonlinear and switching models of the power converters.

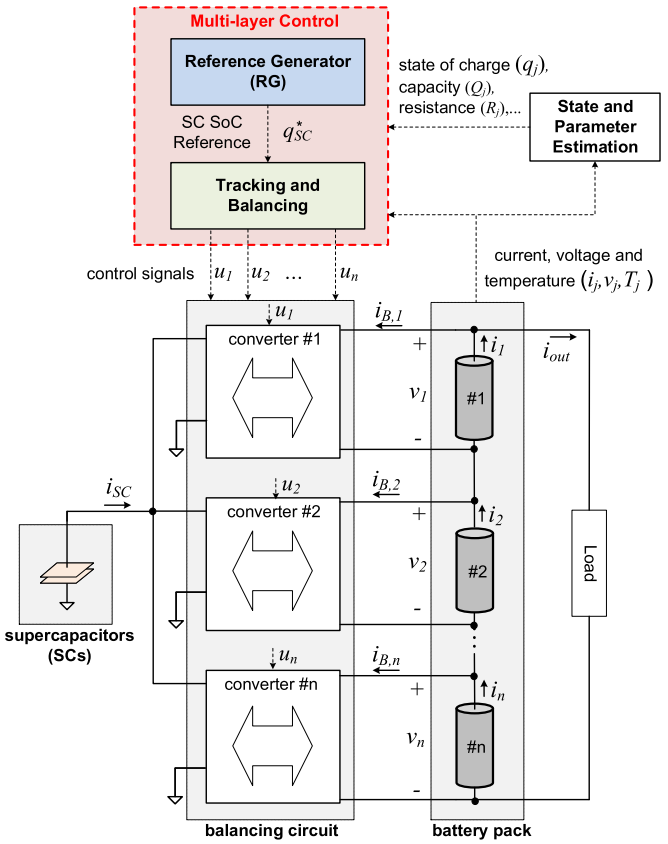


Fig. 1. Block diagram of the HBS and the ML controller, highlighted in red and the main focus of this work.

The inner loops also promote modular solutions: the design of the higher control layers becomes independent of the power conversion, which facilitates the reuse of HBS control algorithms. More specifically, the balancing circuit can be implemented with different types of power conversion (such as buck-boost [29], dual active bridge [30], dual half bridges [16], and among others; see [7], [8] for additional details) while keeping the high-level HBS control unchanged. In addition to control, the HBS also requires diagnosis functions. This includes monitoring of the current, voltage, and temperature of the battery modules, $\{(i_j, v_j, T_j)\}_{j=1}^{n}$ and estimation of relevant states for the HBS operation, such as SoC of the battery modules $\mathbf{q} = \begin{bmatrix} q_1 & \dots & q_n \end{bmatrix}^T$ and their capacity [31].

Our goal is to design a high-level control algorithm for the HBS that fulfills multiple balancing, battery stress, and energy loss goals (to be defined shortly) through manipulation of the reference balancing currents $\mathbf{u}$. The remainder of this section presents a control-oriented model for the HBS, which is then used to specify the control problem in more detail.

B. HBS Model

The HBS model is divided into three main subsystems: 1) the charge and temperature dynamics of the battery modules; 2) the charge dynamics of the SCs; and 3) the power flow between the battery pack, SCs, and the load. To simplify notation, variables related to battery modules and balancing circuits are grouped into n-dimensional vectors, such as $\mathbf{q} \in \mathbb{R}^n$ and $\mathbf{i}_B \in \mathbb{R}^n$.

TABLE I
SUMMARY OF VARIABLES/FUNCTIONS EMPLOYED IN THE HBS MODEL

Sub-Model	Variable	Value
state of charge	$\mathbf{B}_q$	$-\operatorname{diag}\left(\frac{1}{Q_1}, \dots, \frac{1}{Q_n}\right)$
temperature	$\mathbf{A}_T$	$-\operatorname{diag}\left(\frac{1}{C_{h,1}R_{cov,1}}, \dots, \frac{1}{C_{h,n}R_{cov,n}}\right)$
	$\mathbf{B}_T$	$\operatorname{diag}\left(\frac{R_1}{C_{h,1}}, \dots, \frac{R_n}{C_{h,n}}\right)$
	$\mathbf{g}(\mathbf{i})$	$\begin{bmatrix} i_1^2 & \dots & i_n^2 \end{bmatrix}^T$
	$Q_{c,j}(\mathbf{T})$	$(2T_j - T_{j+1} - T_{j-1})/R_{cnd,j}$
i_B control	$\mathbf{A}_B$	$-\operatorname{diag}(\tau_{B,1}^{-1}, \dots, \tau_{B,n}^{-1})$
	$\mathbf{B}_B$	$\operatorname{diag}(\tau_{B,1}^{-1}, \dots, \tau_{B,n}^{-1})$
bal. resistance	$\mathbf{R}_B$	$\operatorname{diag}(R_{b,1} \dots R_{b,n})$
bat. resistance	$\mathbf{R}$	$\operatorname{diag}(R_1, \dots, R_n)$
open-circuit v.	$\mathbf{v}_{oc}(\mathbf{q})$	$\begin{bmatrix} v_{oc,1}(q_1) \dots v_{oc,n}(q_n) \end{bmatrix}^T$
1-vector	$\mathbf{1}$	$\begin{bmatrix} 1 \dots 1 \end{bmatrix}^T$

The first part of the HBS model is defined by the following differential-algebraic representation:

$$\dot{\mathbf{q}} = \mathbf{B}_q \mathbf{i} \tag{1a}$$
$$\dot{\mathbf{T}} = \mathbf{A}_T(\mathbf{T} - \mathbf{1}T_{\text{env}}) + \mathbf{B}_T \mathbf{g}(\mathbf{i}) + \mathbf{Q}_c(\mathbf{T}) \tag{1b}$$
$$\dot{\mathbf{i}}_B = \mathbf{A}_B \mathbf{i}_B + \mathbf{B}_B \mathbf{u} \tag{1c}$$
$$\mathbf{i} = \mathbf{i}_B + \mathbf{1} i_{\text{out}}. \tag{1d}$$

The first equation represents the SoC dynamics of the battery modules, which is dependent on the module currents $\mathbf{i} = \begin{bmatrix} i_1 & \dots & i_n \end{bmatrix}^T$ and a diagonal matrix $\mathbf{B}_q = \operatorname{diag}[-(1/Q_1), \dots, -(1/Q_n)]$, inversely proportional to the nominal capacity Q_j. The SoC $\mathbf{q}$ is further subject to the following safety limits:

$$\underline{\mathbf{q}} \leq \mathbf{q} \leq \overline{\mathbf{q}} \tag{2}$$

where $\underline{\mathbf{q}}$ and $\overline{\mathbf{q}}$ represent the minimum and maximum SoC for the safe operation of the battery module.

Equation (1b) captures the temperature of the battery modules, $\mathbf{T} = \begin{bmatrix} T_1 & \dots & T_n \end{bmatrix}^T$. Similar to [16], we assume a lumped thermal capacitance $(C_{h,j})$ and three main methods for heat transfer. The first is the heat generated by the internal power losses of the battery modules, which are assumed to be dominated by Joule losses of the internal module resistance (R_j). This is captured in (1b) by the term dependent on the quadratic function $\mathbf{g}(\mathbf{i}) = \begin{bmatrix} i_1^2 & \dots & i_n^2 \end{bmatrix}^T$. The second method is related to convective heat transfer between the battery modules and the environment (T_{env}), with convective thermal resistance $R_{\text{cov},j}$. This is captured by the term $\mathbf{A}_T(\mathbf{T} - \mathbf{1}T_{\text{env}})$, where $\mathbf{1} \in \mathbb{R}^n$ is a column vector of ones. The third method deals with conductive heat flow $\mathbf{Q}_c(\mathbf{T}) = [Q_{c,1}(\mathbf{T}), \dots, Q_{c,n}(\mathbf{T})]^T$ between neighboring modules. Similar to [32], we assume that this heat flow depends on the temperature difference between neighboring modules and their thermal resistance $(R_{\text{cnd},j})$, i.e., $Q_{c,j}(\mathbf{T}) = (2T_j - T_{j+1} - T_{j-1})/R_{\text{cnd},j}$. Table I summarizes the values of the matrices $(\mathbf{A}_T, \mathbf{B}_T)$ and functions $(\mathbf{g}, \mathbf{Q}_c)$ employed in the thermal model.

Equation (1c) represents the response of the inner current loops, approximated here by a first-order model $\tau_{B,j}(d/dt)i_{B,j} = -i_{B,j} + u_j$, with dominant time constant $\tau_{B,j}$. This response is encapsulated into a linear state-space representation, characterized by the diagonal matrices $\mathbf{A}_B = -\operatorname{diag}(\tau_{B,1}^{-1}, \dots, \tau_{B,n}^{-1})$, $\mathbf{B}_B = -\mathbf{A}_B$. The current references $\mathbf{u}$ are subject to the following

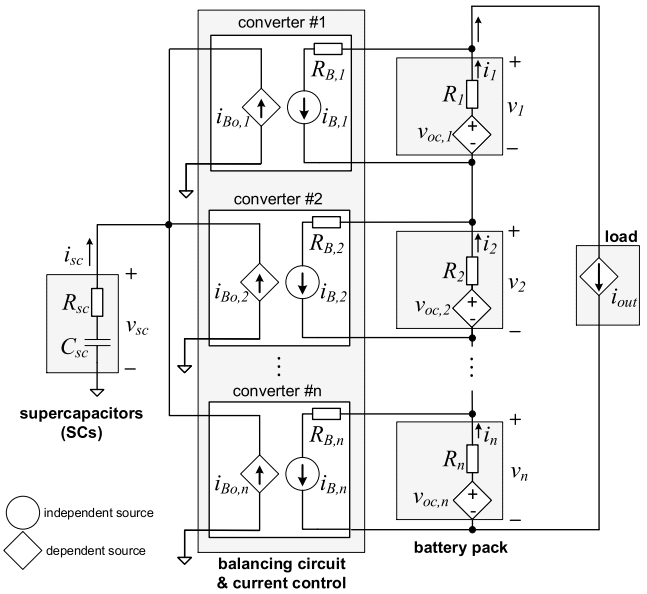


Fig. 2. Equivalent electric circuit of the HBS (module level). The balancing current $i_{B,j}$ in the converters' primary side is independently controlled by the converter's current loop. The currents in the converters' secondary side $i_{\mathrm{Bo},j}$ depend on the power-balance constraint $i_{\mathrm{Bo},j} = (\upsilon_j i_{B,j} - R_{B,j} i_{B,j}^2)/\upsilon_{\mathrm{sc}}$.

physical actuation limits:

$$\mathcal{U} = \{\mathbf{u} \in \mathbb{R}^n : \underline{\mathbf{u}} \leq \mathbf{u} \leq \overline{\mathbf{u}}\} \tag{3}$$

where $\underline{\mathbf{u}}$ and $\overline{\mathbf{u}}$ represent the minimum and maximum balancing currents allowed by the power converters, respectively. The algebraic constraint (1d) defines the current $\mathbf{i}$ applied to the battery modules. This depends on the load (i_{out}) and balancing ($\mathbf{i}_B$) currents and results from the direct application of Kirchhoff's current law.

The second part of the HBS model deals with the SCs. To reduce model complexity, we assume that the SCs response can be approximated by a first-order equivalent circuit [33], composed of an ideal capacitor (C_{sc}) and an internal series resistance (R_{sc}), see Fig. 2. The SoC dynamics of the SCs q_{sc} and voltage υ_{sc} are defined as

$$\dot{q}_{\mathrm{sc}} = -Q_{\mathrm{sc}}^{-1} i_{\mathrm{sc}} \tag{4}$$

$$\upsilon_{\mathrm{sc}} = k_{\mathrm{sc}} q_{\mathrm{sc}} - R_{\mathrm{sc}} i_{\mathrm{sc}} \tag{5}$$

where Q_{sc} is the capacity, i_{sc} the current of the SCs, and $k_{\mathrm{sc}} = Q_{\mathrm{sc}}/C_{\mathrm{sc}}$ is an SoC-to-voltage constant.

To prevent physical damage to the SCs, q_{sc} is subject to the following minimum ($\underline{q}_{\mathrm{sc}}$) and maximum ($\overline{q}_{\mathrm{sc}}$) operational limits:

$$\underline{q}_{\mathrm{sc}} \leq q_{\mathrm{sc}} \leq \overline{q}_{\mathrm{sc}}. \tag{6}$$

The third part of the HBS model captures the power flow between the battery, SCs, and the load. Since the load and battery pack are connected in parallel, the load power (p_{out}) is the product of the load current (i_{out}) and the terminal voltage of the battery pack ($\sum_j \upsilon_j$)

$$p_{\mathrm{out}} = i_{\mathrm{out}} \sum_j \upsilon_j = i_{\mathrm{out}} (\mathbf{1}^{\mathrm{T}} \mathbf{v}) \tag{7a}$$

$$\mathbf{v} = \mathbf{v}_{\mathrm{oc}}(\mathbf{q}) - \mathbf{R}\mathbf{i}. \tag{7b}$$

To decrease model complexity, we assume that the terminal voltage of the battery modules $\mathbf{v} = \left[\upsilon_1, \ldots, \upsilon_n\right]^T$ can be approximated by an SoC-dependent open-circuit voltage $\mathbf{v}_{\mathrm{oc}}(\mathbf{q})$ minus the voltage drop in the internal resistance of the battery modules, $\mathbf{R}\mathbf{i}$, where $\mathbf{R} = \mathrm{diag}(R_1, \ldots, R_n)$ is a diagonal matrix dependent on the internal resistance R_j of the modules.

The power transferred between the power converters and the SCs is defined as

$$p_{\mathrm{sc}} = \mathbf{v}^{\mathrm{T}} \mathbf{i}_B - \mathbf{i}_B^{\mathrm{T}} \mathbf{R}_B \mathbf{i}_B \tag{8a}$$

$$= -k_{\mathrm{sc}} q_{\mathrm{sc}} i_{\mathrm{sc}} + R_{\mathrm{sc}} i_{\mathrm{sc}}^2. \tag{8b}$$

The first algebraic relation (8a) establishes that the power extracted from the battery modules ($\mathbf{v}^{\mathrm{T}}\mathbf{i}_B$) is transferred to the SCs (p_{sc}) and supplies the energy losses of the power converter. These losses are assumed to be lumped into a Joule heating term, $\mathbf{i}_B^{\mathrm{T}}\mathbf{R}_B\mathbf{i}_B$, where $\mathbf{R}_B = \mathrm{diag}(R_{B,1}, \ldots, R_{B,n})$ is a diagonal matrix dependent on the converter's internal resistance $R_{B,j}$. The second algebraic relation (8b) establishes the internal power balance of the SCs.

To simplify the controller design, we assume that the dynamics of the inner current loops (1c) can be neglected, i.e., $\mathbf{i}_B \approx \mathbf{u}$. This simplification is justified due to the time-scale separation that exists between the SoC and temperature dynamics (order of seconds/minutes) and the low-level current loops (order of milliseconds). Another simplification that we adopt is related to the substitution of variables [e.g., (1d) can be directly inserted into (1a) and (1b)], which allow us to reduce the number of algebraic constraints. Due to these simplifications, the HBS model can be compactly represented as

$$\dot{\mathbf{q}} = \mathbf{B}_q(\mathbf{u} + \mathbf{1} i_{\mathrm{out}}) \tag{9a}$$

$$\dot{\mathbf{T}} = \mathbf{A}_T(\mathbf{T} - \mathbf{1} T_{\mathrm{env}}) + \mathbf{B}_T \mathbf{g}(\mathbf{u} + \mathbf{1} i_{\mathrm{out}}) + \mathbf{Q}_c(\mathbf{T}) \tag{9b}$$

$$\dot{q}_{\mathrm{sc}} = -Q_{\mathrm{sc}}^{-1} i_{\mathrm{sc}} \tag{9c}$$

$$0 = h_b(\mathbf{q}, \mathbf{u}, i_{\mathrm{sc}}, q_{\mathrm{sc}}, i_{\mathrm{out}}) = k_{\mathrm{sc}} q_{\mathrm{sc}} i_{\mathrm{sc}} - R_{\mathrm{sc}} i_{\mathrm{sc}}^2 - \mathbf{u}^{\mathrm{T}} \mathbf{R}_B \mathbf{u} + \mathbf{u}^{\mathrm{T}} (\mathbf{v}_{\mathrm{oc}}(\mathbf{q}) - \mathbf{R}(\mathbf{u} + \mathbf{1} i_{\mathrm{out}})) \tag{9d}$$

$$0 = h_o(\mathbf{q}, \mathbf{u}, i_{\mathrm{out}}, p_{\mathrm{out}}) = i_{\mathrm{out}} (\mathbf{1}^{\mathrm{T}} (\mathbf{v}_{\mathrm{oc}}(\mathbf{q}) - \mathbf{R}(\mathbf{u} + \mathbf{1} i_{\mathrm{out}}))) - p_{\mathrm{out}}. \tag{9e}$$

The first three equations capture the SoC and temperature dynamics. The algebraic constraint $h_b(\cdot) = 0$ is related to the power balance between battery modules and SCs, while $h_o(\cdot) = 0$ captures the power balance with the load.

The parameters of HBS model employed in this work are shown in Table II. They are based on a small-scale experimental setup, described in detail in Section V.

C. Control Goals

The HBS control has three goals: 1) equalization of charge and temperature; 2) minimization of energy losses; and 3) minimization of battery stress.

1) Equalization: To fulfill the first goal of charge equalization, it is desirable to operate with small SoC variations between the battery modules [20]. These variations are usually defined as differences with respect to the average behavior of

the battery pack, i.e., $\Delta q_i = q_i - \tilde{q}$, where $\tilde{q} = (1/n)\mathbf{1}^{\mathrm{T}}\mathbf{q}$ is the average SoC. This is equivalent to

$$\Delta\mathbf{q} = \left(\mathbf{I} - \frac{1}{n}\begin{bmatrix}\mathbf{1}^{\mathrm{T}}\\ \vdots \\ \mathbf{1}^{\mathrm{T}}\end{bmatrix}\right)\mathbf{q} = \mathbf{M}\mathbf{q} \tag{10}$$

where $\Delta\mathbf{q} = \left[\Delta q_1, \ldots, \Delta q_n\right]^{\mathrm{T}}$. Similarly, to achieve thermal equalization, it is desirable to operate with small temperature variations $\Delta T_i = T_i - \tilde{T}$, where $\tilde{T} = (1/n)\mathbf{1}^{\mathrm{T}}\mathbf{T}$ is the average temperature in the pack. This is equivalent to

$$\Delta\mathbf{T} = \mathbf{M}\mathbf{T} \tag{11}$$

where $\Delta\mathbf{T} = \left[\Delta T_1, \ldots, \Delta T_n\right]^{\mathrm{T}}$.

2) Losses: The second goal considers power losses (p_l) in the SCs, batteries, and power converters. These losses are compactly defined as

$$\begin{aligned} p_l &= h_l(\mathbf{u}, i_{\mathrm{sc}}, i_{\mathrm{out}}) \\ &= R_{\mathrm{sc}} i_{\mathrm{sc}}^2 + (\mathbf{u} + \mathbf{1}i_{\mathrm{out}})^{\mathrm{T}}\mathbf{R}(\mathbf{u} + \mathbf{1}i_{\mathrm{out}}) + \mathbf{u}^{\mathrm{T}}\mathbf{R}_B\mathbf{u}. \end{aligned} \tag{12}$$

3) Battery Stress and Degradation: Batteries degrade over time (calendar aging) and due to their usage and operation conditions (cycling aging). There are a wide range of physical and chemical mechanisms that contribute to this degradation, including growth of solid electrolyte interphase (SEI), loss of active material, and among others [34].

In this work, we consider the following semiempirical model to capture capacity fade due to cycling aging:

$$Q_j = (1 - \Delta Q_j) Q_{\mathrm{BoL},j} \tag{13}$$

$$\Delta Q_j = \Delta Q_{B,j}(N, I_j, T_j) + \Delta Q_{E,j}(N, I_j, T_j) \tag{14}$$

where Q_j is the module capacity, $Q_{\mathrm{BoL},j}$ the beginning-of-life capacity, N the number of charge/discharge cycles, and I_j the C-rate.[1] The function $\Delta Q_j(.)$ expresses the normalized loss of capacity, composed of two terms. The first term, $\Delta Q_{B,j}$, captures the slow aging mechanisms that occur during the initial battery lifetime and is modeled through an Arrhenius-like formula [35]

$$\Delta Q_{B,j}(\cdot) = \theta_{j,1} \exp\left(-\frac{\theta_{j,4}}{T_j} + \left(\theta_{j,2} + \frac{\theta_{j,5}}{T_j}|I_j|\right)\right) N^{\theta_{j,3}} \tag{15}$$

where $\theta_{j,1}, \ldots, \theta_{j,5}$ are parameters of the model. The second term, $\Delta Q_{E,j}$ captures fast aging mechanisms that happen later in the battery lifetime

$$\Delta Q_{E,j}(\cdot) = \theta_{j,8} \exp(N - N_{j,\mathrm{knee}}(\cdot)\theta_{j,7}) \tag{16}$$

$$N_{j,\mathrm{knee}}(\cdot) = \theta_{j,6} + \theta_{j,9} I_j + \theta_{j,10} T_j \tag{17}$$

where $\theta_{j,6}, \ldots, \theta_{j,10}$ are parameters, and $N_{j,\mathrm{knee}}(\cdot)$ is the "knee point" where the transition to fast aging occurs. Fig. 3(a) depicts the trend of this model, parameterized for lithium nickel manganese cobalt (NMC)-graphite cells [36] similar to the ones employed in our experimental setup. The results show a loss of capacity as N increases, which becomes more severe at higher temperatures (T) and currents (I).

[1]C-rate is a normalized current metric, expressed as the ratio between current i_j and the nominal capacity (in ampere-hours).

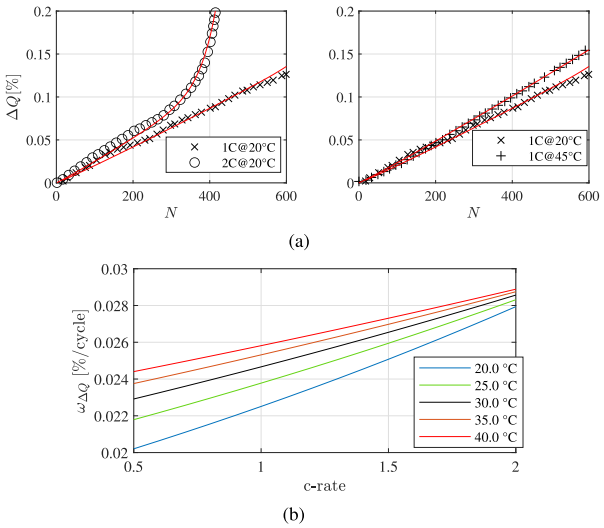


Fig. 3. (a) Normalized capacity drop (ΔQ) of the battery modules for different temperatures and currents; red lines indicate the fitting of the model (14) using experimental data ($\times$, $+$, $\circ$) from [36]; aging model parameters included in the Appendix. (b) Effect of temperature and current (c-rate) in the aging rate ($\omega_{\Delta Q}$), evaluated at $N_0 = 300$.

In addition to capacity loss, it is also useful to consider the aging rate $\omega_{j,\Delta Q}$ of the battery module, i.e., how fast the capacity drops over the cycles. This can be obtained by evaluating the rate of change of ΔQ_j at a given cycle number N_0

$$\omega_{j,\Delta Q}(I_j, T_j) = \left.\frac{d\Delta Q(N, I_j, T_j)}{dN}\right|_{N=N_0}. \tag{18}$$

Ideally, the aging rate $\omega_{j,\Delta Q}$ should be minimized in order to extend battery lifetime. However, this brings two difficulties. First, this aging model is nonlinear, which complicates the controller design, especially in predictive and optimization-based frameworks as investigated in this work. Second, $\omega_{j,\Delta Q}$ depends on a large number of parameters $(\theta_{j,1}, \ldots, \theta_{j,10})$, $j = 1, \ldots, n$, which considerably increase calibration effort if $\omega_{j,\Delta Q}$ is explicitly included in the controller design. To overcome these challenges, we adopt here an implicit approach: we minimize battery stress factors (current and temperature) as proxies for aging rate $\omega_{j,\Delta Q}$. This simplification is justified by the strong correlation that exists between the value of these stress factors and $\omega_{j,\Delta Q}$ for the battery operating window considered in this work [see Fig. 3(b)]. Another simplification that we follow here is related to temperature. In the sequel, we minimize heat dissipation in the battery as a mean to decrease temperature (and degradation). Leveraging these simplifications, we seek to reduce battery degradation via penalization of current and dissipated heat, which is encoded into the following cost functions:

$$J_{sI} = \sum_j i_j^2, \quad J_{sT} = \sum_j Q_{\mathrm{heat},j} \tag{19}$$

where $Q_{\mathrm{heat},j}$ is the heat generated by the battery module j. Since the heat is assumed to be proportional to the square of the module current ($Q_{\mathrm{heat},j} = R_j i_j^2$), both costs can be grouped together

$$J_S = \sum_j \alpha_{S,j} i_j^2 = \mathbf{i}^{\mathrm{T}}\boldsymbol{\alpha}_S\mathbf{i} = (\mathbf{u} + \mathbf{1}\mathbf{i}_{\mathrm{out}})^{\mathrm{T}}\boldsymbol{\alpha}_S(\mathbf{u} + \mathbf{1}\mathbf{i}_{\mathrm{out}}) \tag{20}$$

TABLE II
PARAMETERS OF THE HBS MODEL

Variable	Symbol	Value	Unit
Battery pack			
nominal capacity	Q_j	$[3,3,4]\times 3$	A·h
internal resistance	R_j	$[\frac{1}{3},\frac{1}{3},\frac{1}{4}\ ]\times 51.16$	mΩ
open-circuit voltage	$v_{oc,j}(q_j)$	$3.31+0.88q_j$	V
thermal capacitance	$C_{h,j}$	$[145,145,191]$	J/K
convective resistance	$R_{cov,j}$	$[10.7,10.7,7.3]$	K/W
conductive resistance	$R_{cnd,j}$	∞	K/W
initial SoC	$\mathbf{q}(\mathbf{0})$	$[1,1,1]^T$	-
initial temperature	$\mathbf{T}(\mathbf{0})$	$[1,1,1]^T\times T_{env}$	°C
env. temperature	T_{env}	25	°C
weight for stress factor	$\alpha_{S,j}$	$[(\frac{4}{3})^2,(\frac{4}{3})^2,(\frac{4}{4})^2]$	-
SoC limits	$\underline{\mathbf{q}}$	$[0.2,0.2,0.2]^T$	-
	$\overline{\mathbf{q}}$	$[1,1,1]^T$	-
Supercapacitors			
capacitance	C_{sc}	496	F
nominal capacity	Q_{sc}	0.77	A·h
internal resistance	R_{sc}	1.4	mΩ
SoC-voltage gain	k_{sc}	5.4	V
SoC limits	$(\underline{q}_{sc},\overline{q}_{sc})$	(0.5,1)	-
initial SoC	$q_{sc}(0)$	1.0	-
Balancing circuit			
inner resistance	$R_{B,j}$	0.04	Ω
balancing current limits	$\mathbf{u}_{min}$	$-[5,5,5]^T$	A
	$\mathbf{u}_{max}$	$[5,5,5]^T$	A

where $\boldsymbol{\alpha}_S = \text{diag}(\alpha_{S,1},\ldots,\alpha_{S,n})$ is a stress weight matrix. The scalar stress weight $\alpha_{S,j}$ is inversely dependent on the module's present capacity, i.e., $\alpha_{S,j}(Q_j) = (Q_{\text{BoL},j}/Q_j)^2$. The resulting weighting strategy assigns a higher stress penalty to weaker modules (i.e., that experienced higher capacity fade), protecting them from additional stress and accelerated aging. A final note on the battery aging model; although not explicitly included in the controller design, the battery aging rate $\omega_{j,\Delta Q}$ is used to evaluate controller performance in Section V.

D. Control Problem

The main goal of this work can now be formulated as a multiobjective constrained optimization problem.

Problem 1: Find the balancing currents $\mathbf{u} \in \mathcal{U}$ that minimize the following.

1) *(q) SoC Imbalances:* $\|\Delta\mathbf{q}\| \to 0$.
2) *(T) Thermal Imbalances:* $\|\Delta\mathbf{T}\| \to 0$.
3) *(S) Battery Stress:* $J_S \to 0$.
4) *(L) Power Losses:* $p_l \to 0$

while fulfilling the power-balance constraints (9d) and (9e) and SoC limits (2) and (6).

III. SINGLE-LAYER QTSL

Problem 1 represents a challenging control problem. It has a potentially large number of decision variables (n), with physical constraints and nonlinearities. It also has a multitude of objectives; the control input is subject to algebraic constraints (9d) and (9e), which confine $\mathbf{u}$ to a manifold dependent on p_{out} and other variables. To simultaneously cope with all these challenges, we design in this section an SL controller, where all performance goals are addressed in the same MPC formulation. In the sequel, we call this approach SL *qTSL* since it allows us to simultaneously manage charge (q), temperature (T), stress (S), and losses (L).

A. Problem Formulation

Let us first discretize the dynamics from (9) using Euler's method and sample time τ_s

$$\begin{aligned}
\mathbf{q}(k+1) &= \mathbf{f}_q(k) = \mathbf{q}(k) + \tau_s\tilde{\mathbf{f}}_q(\mathbf{u}(k), i_{\text{out}}(k))\\
\mathbf{T}(k+1) &= \mathbf{f}_T(k) = \mathbf{T}(k) + \tau_s\tilde{\mathbf{f}}_T(\mathbf{T}(k), \mathbf{u}(k), T_{\text{env}}(k), i_{\text{out}}(k))\\
q_{\text{sc}}(k+1) &= f_{\text{sc}}(k) = q_{\text{sc}}(k) - \tau_s Q_{\text{sc}}^{-1} i_{\text{sc}}(k)
\end{aligned}$$

where k is the discrete time index and $\mathbf{f}_T$, $\mathbf{f}_q$, and f_{sc} are vector fields obtained from (9a)–(9c), respectively.

The SL *qTSL* framework uses the above discrete model to predict the response of the system over an N-step horizon. It maps the control goals, defined in Problem 1, to the following optimization problem:

$$\min_{\mathbf{u}(l|k)} \quad J_{qTSL} = \sum_{l=0}^{N}\Big(\alpha_q J_q(l|k) + \alpha_T J_T(l|k)\Big) \tag{22a}$$
$$+\alpha_{\text{SL}}\sum_{l=0}^{N-1}\Big(\beta J_S(l|k) + (1-\beta)J_L(l|k)\Big)$$
$$\text{s.t.} \quad \mathbf{q}(l+1|k) = \mathbf{f}_q(l|k), \tag{22b}$$
$$\mathbf{T}(l+1|k) = \mathbf{f}_T(l|k), \tag{22c}$$
$$q_{\text{sc}}(l+1|k) = f_{\text{sc}}(l|k) \tag{22d}$$
$$0 = h_b(\mathbf{q}(l|k), \mathbf{u}(l|k), i_{\text{sc}}(l|k), q_{\text{sc}}(l|k), i_{\text{out}}(l|k)) \tag{22e}$$
$$0 = h_o(\mathbf{q}(l|k), \mathbf{u}(l|k), i_{\text{out}}(l|k), p_{\text{out}}(l|k)) \tag{22f}$$
$$\mathbf{u}(l|k) \in \mathcal{U}, \tag{22g}$$
$$\underline{\mathbf{q}} \le \mathbf{q}(l+1|k) \le \overline{\mathbf{q}}, \quad \underline{q}_{\text{sc}} \le q_{\text{sc}}(l+1|k) \le \overline{q}_{\text{sc}} \tag{22h}$$
$$\mathbf{q}(0|k) = \mathbf{q}(k), \quad \mathbf{T}(0|k) = \mathbf{T}(k), \quad q_{\text{sc}}(0|k) = q_{\text{sc}}(k)$$
$$l = 0,\ldots,N-1 \tag{22i}$$

where $(l|k)$ indicates the predicted value of a given variable at the time instant $k+l$.

The above problem has four cost functions

$$\begin{aligned}
J_q(l|k) &= \Delta\mathbf{q}(l|k)^{\text{T}}\Delta\mathbf{q}(l|k) = \mathbf{q}(l|k)^{\text{T}}\mathbf{M}^{\text{T}}\mathbf{M}\mathbf{q}(l|k)\\
J_T(l|k) &= \Delta\mathbf{T}(l|k)^{\text{T}}\Delta\mathbf{T}(l|k) = \mathbf{T}(l|k)^{\text{T}}\mathbf{M}^{\text{T}}\mathbf{M}\mathbf{T}(l|k)\\
J_S(l|k) &= (\mathbf{u}(l|k) + \mathbf{1}\mathbf{i}_{\text{out}}(l|k))^{\text{T}}\boldsymbol{\alpha}_S(\mathbf{u}(l|k) + \mathbf{1}\mathbf{i}_{\text{out}}(l|k))\\
J_L(l|k) &= p_l(l|k) = h_l(\mathbf{u}(l|k), i_{\text{sc}}(l|k), i_{\text{out}}(l|k)).
\end{aligned}$$

The first two costs penalize SoC and thermal imbalances over the prediction horizon, the third penalizes battery stress, and the last term penalizes energy losses. All four costs are quadratic in the decision variables, while α_q, α_T, α_{SL}, and β are tuning weights, selected by the designer.

In addition to the discrete HBS model (22b)–(22f), the prediction model also contains the physical limits (22g) and (22h) and initial conditions (22i). The nonlinearities in the prediction model are mainly due to quadratic terms dependent on the currents, which appear, e.g., in the temperature dynamics and power-balance constraints. It is also worth pointing out that, because of the augmentation with the initial state, the predicted SoCs and temperature states $[\mathbf{q}(l|k), \mathbf{T}(l|k), q_{\text{sc}}(l|k)]$ and related cost functions $[J_q(l|k),\ J_T(l|k)]$ have a longer index range $l \in [0, N]$ than the control inputs ($\mathbf{u}(l|k) \in [0, N-1]$) and their associated costs $[J_S(l|k),\ J_L(l|k)]$.

The variables $p_{\text{out}}(l|k), T_{\text{env}}(l|k)\ \ l = 0,\ldots,N$ are treated here as exogenous inputs. There are two approaches to define

the value of these exogenous inputs during the prediction horizon. The first approach exploits preview information, e.g., based on the forecast of the output power, which leads to a *non-causal qTSL*. Note that, with recent advances in vehicle-to-infrastructure communications and the collection of real-time traffic data, as well as increase in vehicle automation, load forecast of vehicles is becoming easier to provide [37]. The second approach does not have access to preview information. It assumes a predefined behavior for the exogenous inputs during the prediction horizon such that a *causal qTSL* can be obtained, for example, constant behavior

$$p_{\text{out}}(l|k) = p_{\text{out}}(k), \quad T_{\text{env}}(l|k) = T_{\text{env}}(k) \tag{23}$$

for $l = 0, \ldots, N$, where $[p_{\text{out}}(k), T_{\text{env}}(k)]$ represent the load power and environmental temperature at the current time instant.

The dynamics of the prediction model in (22) are inherent slow in time. The SoC dynamics (22b) and (22d) are based on the integration of the battery and SCs current, while the temperature dynamics have a dominant time constant $(C_{h,j} R_{\text{cov},j})$ on the order of minutes. In practice, the load power disturbance p_{out} represents the fastest variable, which is typically discretized with a 1-s sample time [38]. Because of this, a sample time $\tau_s = 1$ s is adopted in the discretization of the prediction model.

The *qTSL* MPC implementation follows a receding horizon strategy, i.e., at each time step, the optimal solution of (22) is computed $[\mathbf{u}^*(l|k)]$, and then, the first element $[\mathbf{u}^*(0|k)]$ applied to the HBS. Because of nonlinear constraints, e.g., (22f), finding optimal solutions is a challenging task. To assist in the search of optimal solutions, an interior point optimizer (IPOPT) [39] was employed.

B. Analysis and Tuning Guidelines

The SL *qSTL* framework allows us to address all control goals established in Problem 1. However, this also brings several challenges. One challenge is related to the MPC calibration, i.e., how to select the weights α_q, α_T, α_{SL}, and β, as well as prediction horizon (N), such that all costs are simultaneously minimized. Computational effort is another challenge; selecting a large prediction horizon (N) might improve performance, but the computational cost increases, compromising real-time execution. In addition, load forecast (23) might not be always available across long horizons, which might also have a negative impact on the control performance. These issues are analyzed in this section.

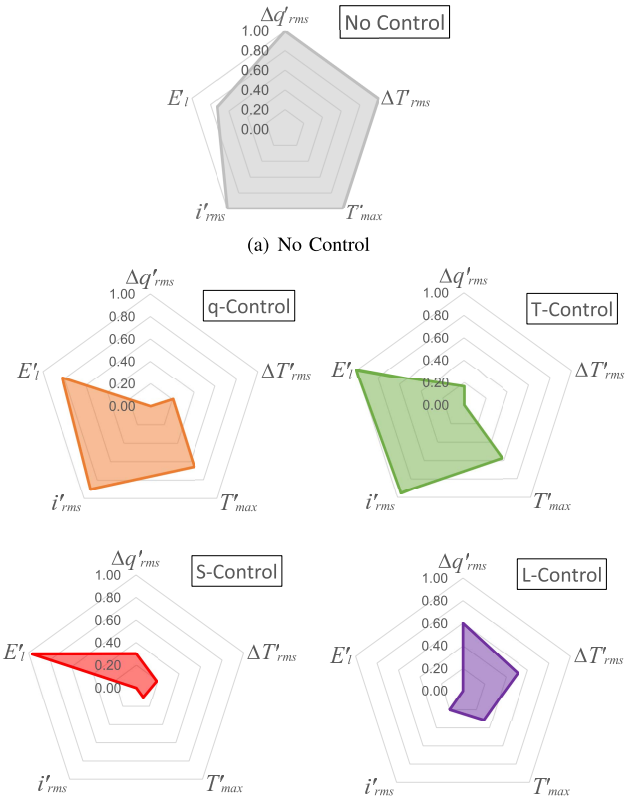
To assist us in the analysis, we define several performance metrics, related to the SoC imbalances (Δq_{rms}), temperature imbalances (ΔT_{rms}), energy losses (E_l), and battery stress metrics, such as root-mean-square current i_{rms} and maximum temperature T_{max}. Table III presents the definitions of these metrics.

1) Guidelines for Weight Calibration: We first consider mono-objective MPC controllers that focus only on one performance criterion and are parameterized as follows.

1) *q-Controller*: $\alpha_q = 1$, $\alpha_T = 0$, $\alpha_{\text{SL}} = 0$, and $\beta = 0$.
2) *T-Controller*: $\alpha_q = 0$, $\alpha_T = 1$, $\alpha_{\text{SL}} = 0$, and $\beta = 0$.
3) *S-Controller*: $\alpha_q = 0$, $\alpha_T = 0$, $\alpha_{\text{SL}} = 1$, and $\beta = 1$.
4) *L-Controller*: $\alpha_q = 0$, $\alpha_T = 0$, $\alpha_{\text{SL}} = 1$, and $\beta = 0$.
5) *No Control*: $\mathbf{u} = 0$.

TABLE III
DEFINITION OF PERFORMANCE METRICS

1: average rms value of SoC differences $\Delta q_{rms} = \frac{1}{n}\sum_{j=1}^{n} \text{rms}(\Delta q_j)$
2: average rms value of temperature differences $\Delta T_{rms} = \frac{1}{n}\sum_{j=1}^{n} \text{rms}(\Delta T_j)$
3: maximum temperature in the battery modules $T_{max} = \max_{j,k} T_j(k)$
4: average rms value of the battery current $i_{rms} = \frac{1}{n}\sum_{j=1}^{n} \text{rms}(i_j)$
5: energy losses $E_l = \int_0^{T_{dc}} p_l(t)dt$
T_{dc} = duration of the driving cycle, n = number of modules, $\text{rms}(x) = \sqrt{\frac{1}{T_{dc}}\int_0^{T_{dc}} x^2(t)dt}$



(a) No Control

(b) Mono-Objective Controllers

Fig. 4. Normalized performance metrics for (a) no control and (b) mono-objective controllers ($N = 100$). Note: ideal controller would be a single point at the origin (with zero normalized cost); the chart's area measures the overall distance to such ideal controller.

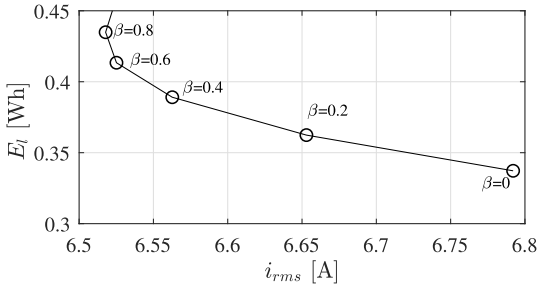


Fig. 5. Tradeoff between battery stress and energy losses during β calibration.

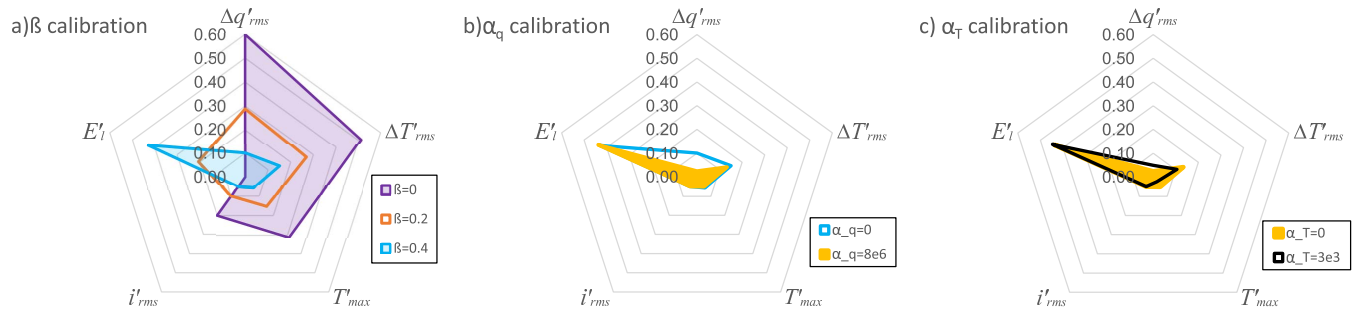


Fig. 6. (a) Calibration of weight β. (b) Calibration of weight α_q (with $\beta = 0.4$). (c) Calibration of weight α_T (with $\beta = 0.4$, $\alpha_q = 8e6$). Final weighted-calibrated *qTSL* controller: ($\alpha_{\text{SL}} = 1$, $\beta = 0.4$, $\alpha_q = 8e6$, $\alpha_T = 3e3$, and $N = 100$).

These mono-objective controllers can be used to determine the best attainable values for the performance metrics, i.e., utopic metric values.

Next, we normalize the performance metrics to facilitate the comparison between controllers

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{24}$$

where x is one of the performance metrics defined in Table III, x' is its normalized value, $x_{\min}$ is the minimum value of the metric x obtained with mono-objective controllers, and $x_{\max}$ is the maximum. In this normalization, x' lies in the range [0, 1]. If $x' = 1$, then the metric is at the maximum cost achieved by all controllers, while a zero value implies the minimum cost.

Fig. 4 shows the normalized performance metrics of the mono-objective controllers obtained with a scaled version of the driving cycle US06 [38], prediction window $N = 100$, and sample time $\tau_s = 1$ s. The results indicate that each mono-objective controller has different strengths and weaknesses. The q- and *T-controllers* are good for balancing purposes, offering the lowest costs for $\Delta q'_{\text{rms}}$ and $\Delta T'_{\text{rms}}$, but exhibiting higher currents and temperatures. The *S-controller* provides good balancing and stress performance, but at the expense of high energy losses. The *L-controller* reduces energy losses but degrades all other metrics.

To tune the MPC weights, we propose the following practical guidelines.

1) *β Calibration:* We initialize the controller with L parameterization ($\alpha_q = 0, \alpha_T = 0$, and $\alpha_{\text{SL}} = 1$) and then search for the weight β that provides an acceptable tradeoff among the battery stress and energy losses criteria, i.e., the knee of the curve shown in Fig. 5.
2) *α_q Calibration:* Next, we move to the tuning of α_q and focus on further reduction of SoC imbalances. Fig. 6(b) shows that higher α_q decreases $\Delta q'_{\text{rms}}$ in the direction of the utopic value set by the *S-controller*.
3) *α_T Calibration:* The final step consists of increasing α_T as a means to further reduce the thermal imbalances [Fig. 6(c)]. This step might also degrade other metrics, particularly SoC balancing; if this occurs, the designer should use α_T as the final "tuning knob" to obtain an acceptable tradeoff between the performance metrics.

Fig. 6 shows the effect of the three main calibration steps. The performance of the calibrated $qTSL$ [Fig. 6(c)] is very close to the best balancing and stress metrics obtained with mono-objective controllers while providing a good comprise with energy losses (in the sense that these losses are still inferior to the ones generated by the *No Control* variant).

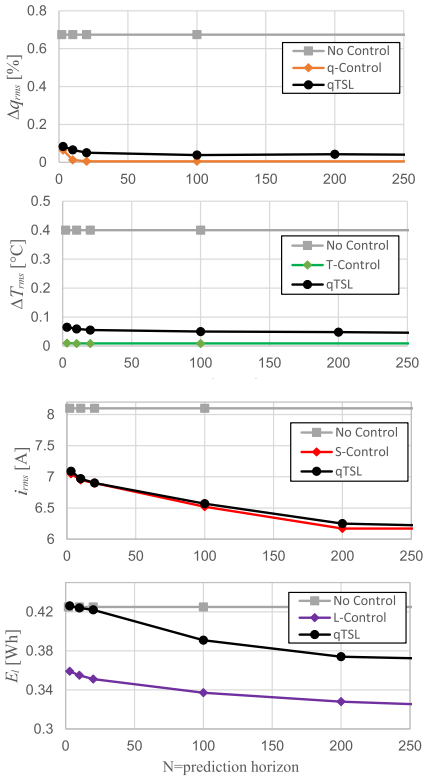


Fig. 7. Effect of the length of the prediction horizon (N) on the performance of mono-objective and SL *qTSL* controllers. Note: longer prediction horizons significantly reduce battery stress (-10%) and energy losses (-20%) but do not affect balancing goals.

2) Prediction Horizon: To investigate the impact of the prediction horizon (N), the mono-objective MPC controllers and the calibrated *qSTL* were evaluated for different values of N (Fig. 7). As expected, the mono-objective controllers provide the best performance for their respective criteria, defining a lower bound for the cost metrics. For SoC and temperature balancing, the mono-objective controllers achieve the minimum costs for a small prediction horizon ($N < 10$), while longer horizons ($N > 200$) are needed to minimize battery stress (i_{rms}) and energy losses (E_l). The calibrated *qTSL* follows a similar trend as the mono-objective controllers: higher prediction horizons N reduce the cost metrics. One can also observe that the gap between *qTSL* and mono-objective controllers is relatively small for most of the metrics; the energy losses are the only exception.

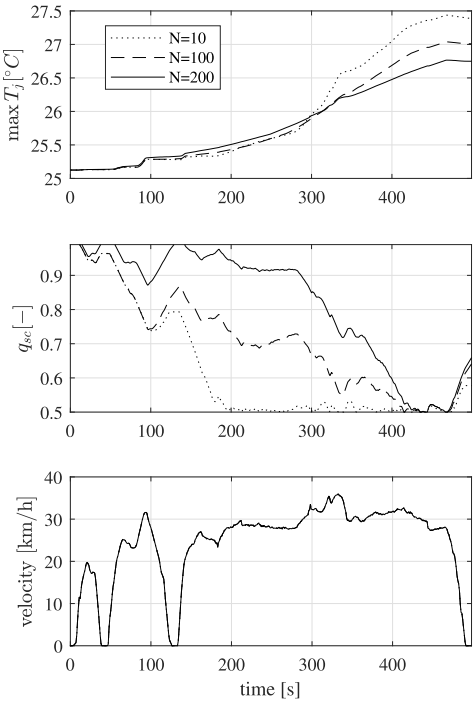


Fig. 8. Effect of prediction horizon N in the maximum battery temperature $[\max_{j\in\{1,\ldots,n\}} T_j(t)]$ and SC SoC for the *S-controller*.

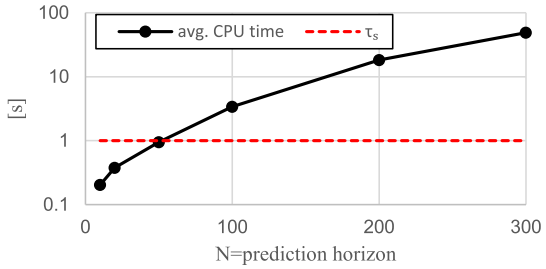


Fig. 9. Effect of the prediction horizons (N) on the average computational time of the SL *qTSL* controller (black line); sample time τ_s depicted by the red-dotted line.

The battery stress is the most sensitive metric to the prediction horizon N. To better understand this issue, Fig. 8 shows the SoC of the SCs, maximum temperature, and vehicle velocity for the *S-controller* with different prediction windows. Shorter prediction horizons quickly discharge the SCs (in less than 200 s) and reduce battery stress only during the first part of the driving cycle. The problem with this strategy is that by the time, the vehicle reaches maximum speed and requires high power levels (around 300–350 s) and the SCs are already discharged and are unable to aid the battery pack. As a result, higher temperatures are observed during the second part of the driving cycle. In contrast, controller parameterizations with longer prediction horizons ($N = 200$) do not discharge the SCs at the beginning of the driving cycle; instead, they strategically deploy the SCs charge during the second part of the driving cycle (300–350 s) to aid the battery during the period with the highest load. As a result, lower temperatures are observed.

This analysis demonstrates that a larger prediction horizon N is key to prepare the SCs for future loads/disturbances and reduce battery stress. However, this comes with a price: higher computational times. This is shown in Fig. 9, which shows

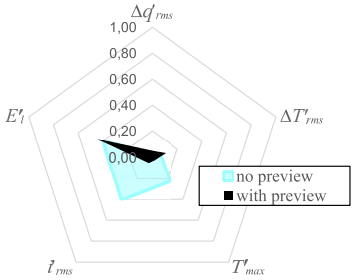


Fig. 10. Effect of preview information in the SL *qTSL* controller ($N = 100$).

the computation time of the SL *qTSL* in the HBS embedded platform based on a Raspberry pi 3 [40]. The computation times reach dozens of seconds when the MPC prediction horizon is larger than 40 s, violating real-time timing constraint associated with the MPC sample time (τ_s).

3) Uncertain Load Forecast: The preview information about the load power, $p_{\text{out}}(k)$, might not be always available. To better understand the impact of this information, Fig. 10 shows the normalized performance of the calibrated *qTSL* controller: 1) with exact preview information and 2) without preview information and using the causal formulation (23). These results show that the SoC and temperature balancing metrics have a weak dependence on the preview information; this is in line with the findings in Section III-B2, which demonstrates that short prediction horizons are enough for balancing tasks. The current and temperature stress are the most sensitive metrics to preview information. Fig. 10 shows a 40% decrease in normalized current and 20% decrease in the normalized temperature when the *qTSL* operates with preview.

IV. Multilayer *qTSL*

Solving the SL MPC described in the previous section is difficult. It requires the fulfillment of multiple goals (q, T, S, and L), involves complex *module-level* models of the battery, and needs long prediction windows to optimize battery stress while complying with computational time constraints. One possibility to fulfill these latter constraints is to shorten the prediction horizon of the SL MPC. As discussed in Section III-B2, this strategy works well for battery balancing tasks, but it induces "greedy" energy management behaviors that focus too much on short-term performance and lead to a quick discharge of SCs (see Fig. 8). This compromises the SCs ability to effectively support the battery pack and alleviate stress during the entire driving cycle. To mitigate this shortcoming, we devise a two-layer MPC that operates on different prediction horizons, model complexities, and time scales (Fig. 11).

The high-level layer, called the reference generator (RG), focuses on long-term energy management planning for the SCs using a slow time scale. This layer computes reference trajectories for the SC SoC (q^*_{sc}) that optimizes battery stress and losses over longer prediction windows; it exploits simplified *pack-level* prediction models and states, discretized with long sample times, to keep computational effort low. The low-level layer handles the short-term energy management and balancing tasks with a higher degree of granularity and *module-level* models using a fast time scale. It is based on the *qTSL*

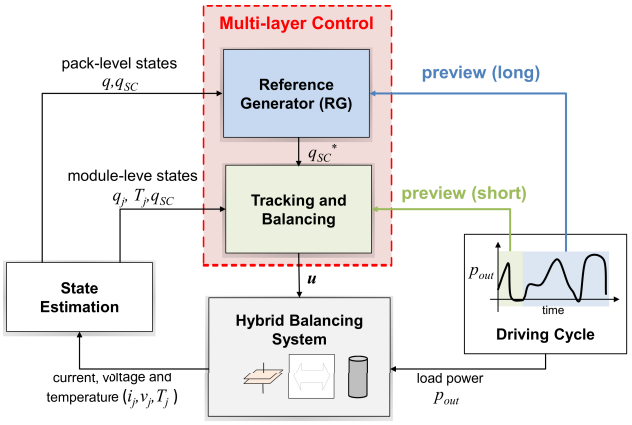


Fig. 11. Block diagram of the ML *qTSL* and its use of different prediction horizons for the load forecast.

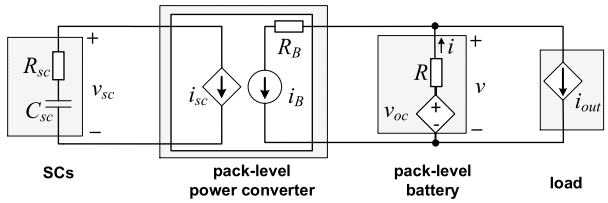


Fig. 12. Pack-level model of the HBS employed in the RG.

TABLE IV
VARIABLES AND PARAMETERS OF THE PACK-LEVEL MODEL OF THE BATTERY AND BALANCING CIRCUITS

Variable	Description
$q = \frac{1}{n}\sum_{j=1}^{n} q_j$	average SoC of the battery modules
$i = \frac{1}{n}\sum_{j=1}^{n} i_j$	average current in the battery modules
$Q = \frac{1}{n}\sum_{j=1}^{n} Q_j$	average battery module capacity
$R = \sum_{j=1}^{n} R_j$	equivalent resistance of the battery
$i_B = \frac{1}{n}\sum_{j=1}^{n} i_{B,j}$	average balancing current
$v_{oc}(q) = \sum_{j=1}^{n} v_{oc,j}(q)$	open-circuit voltage of the battery pack based on average SoC
$v = v_{oc}(q) - Ri$	terminal voltage of the battery pack
$R_B = \sum_{j=1}^{n} R_{B,j}$	equivalent resistance of balancing converters

formulation presented in (22), augmented with an additional cost term to track the reference trajectory q_{sc}^*. The reference q_{sc}^* indicates the most suitable SC charge level that the low-level layer should maintain to fulfill long-term energy management goals. It is a critical mechanism to prevent the low-level MPC (which uses short prediction horizons and complex *module-level* models for balancing purposes) from becoming "greedy" and quickly discharge the SCs.

A. High-Level Layer: RG

The RG prediction model neglects cell-to-cell variations and views the battery pack as one virtual cell and the balancing circuits as one virtual power converter (see Fig. 12). Only average values of the battery SoC (q), current (i), and capacity (Q), as well as average balancing current (i_B), are considered. These average values are then used to compute the open-circuit voltage (v_{oc}), terminal voltage (v), and internal resistance (R) of the battery pack. The resulting *pack-level* battery model can be approximated by an equivalent open-circuit voltage connected in series with an equivalent inner resistance, which is then connected in parallel with the virtual power converter (see Fig. 12). The virtual power converter captures the average

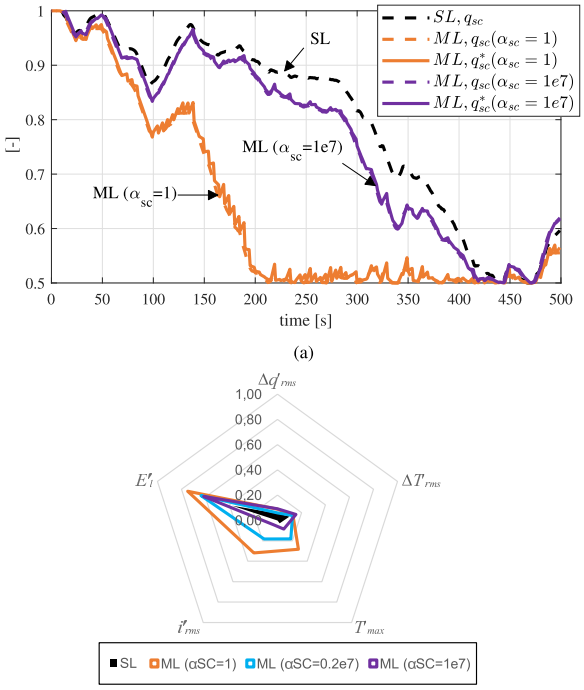


Fig. 13. Simulation results of (a) SL control with $N\tau_s = 200$ s and (b) ML with different SoC tracking weights, $N_R\tau_s^R = 200$ s, $N\tau_s = 10$ s. (b) Performance metrics.

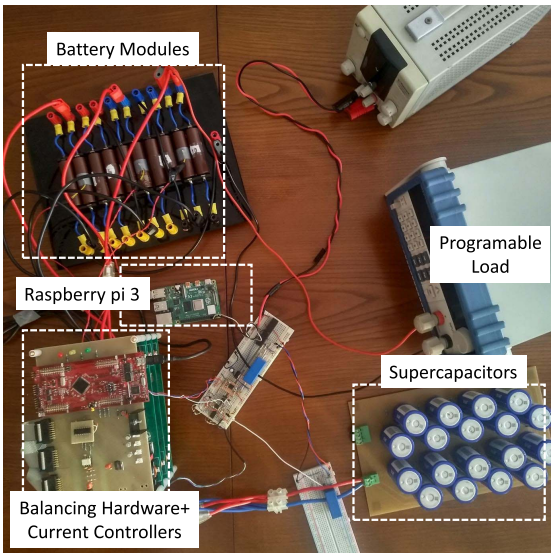


Fig. 14. Small-scale laboratory prototype employed in the experimental validation of the *qTSL* control framework.

pack-level flow of power from the battery and the SCs and is subject to the following power-balance constraint:

$$p_{\text{sc}} = -v_{\text{sc}} i_{\text{sc}} = i_B v - i_B^2 R_B \tag{25}$$

where R_B is the inner resistance of the equivalent power converter. This constraint links the average power extracted from the battery pack ($i_B v$) with the power delivered to the SCs (p_{sc}) and Joule losses in the converter ($i_B^2\ R_B$). Table IV describes the *pack-level* variables and parameters employed in the model.

Next, we combine the *pack-level* model of the battery and power converter together with the SCs and load

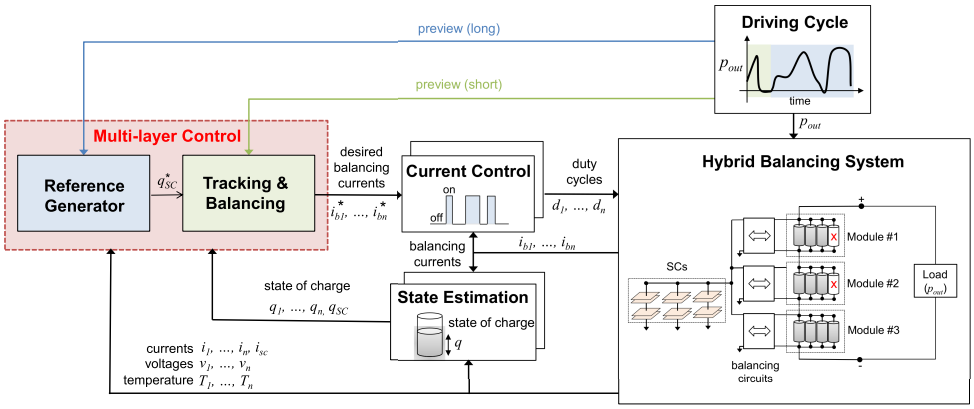


Fig. 15. Block diagram of the setup employed in the experimental validation of the ML *qTSL* control framework. Note: battery modules #1 and #2 each have a faulty cell, which is disconnected from the modules.

models (4) and (7a). This results in the following *pack-level* HBS model:

$$\dot{q} = -\frac{1}{Q} i, \quad \dot{q}_{\text{sc}} = -\frac{1}{Q_{\text{sc}}} i_{\text{sc}} \tag{26a}$$

$$i = i_{\text{out}} + i_B \tag{26b}$$

$$p_{\text{out}} = v i_{\text{out}} = (v_{oc}(q) - Ri) i_{\text{out}} \tag{26c}$$

$$\begin{aligned} 0 &= h_r(q_{\text{sc}}, i_{\text{sc}}, q, i_B) \\ &= k_{\text{sc}} q_{\text{sc}} i_{\text{sc}} + (v_{oc}(q) - Ri) i_B - R_B i_B^2 - R_{\text{sc}} i_{\text{sc}}^2. \end{aligned} \tag{26d}$$

The first set of equations represents the average SoC dynamics of the battery and SC pack. The second equation relates the average current in the battery modules (i) to the output current (i_{out}) and average converter current (i_B). The third and fourth equations represent algebraic power-balance constraints between the load/battery/SCs.

The MPC formulation of the RG uses a discrete-time representation of (IV-A) to predict the response of the system. To further decrease the computational load, the RG controller operates with a slower sample time (τ_s^R). The RG optimization problem is formulated as

$$\min_{i_B(.)} \quad J_R = \sum_{l=0}^{N_R-1} \left(\beta_R i^2(l|k_R) + (1-\beta_R) \tilde{p}(l|k_R) \right)$$

$$\text{s.t.} \quad q(l+1|k_R) = q(l|k_R) - \tau_s^R \frac{1}{Q} i(l|k_R), \tag{27a}$$

$$q_{\text{sc}}(l+1|k_R) = q_{\text{sc}}(l|k_R) - \tau_s^R \frac{1}{Q_{\text{sc}}} i_{\text{sc}}(l|k_R) \tag{27b}$$

$$i(l|k_R) = i_{\text{out}}(l|k_R) + i_B(l|k_R) \tag{27c}$$

$$p_{\text{out}}(l|k_R) = (v_{oc}(q(l|k_R)) - Ri(l|k_R)) i_{\text{out}}(l|k_R)$$

$$h_r(q_{\text{sc}}(l|k_R), i_{\text{sc}}(l|k_R), q(l|k_R), i_B(l|k_R)) = 0$$

$$\underline{i}_B \le i_B(l|k_R) \le \overline{i}_B \tag{27d}$$

$$\underline{q}_{\text{sc}} \le q_{\text{sc}}(l|k_R) \le \overline{q}_{\text{sc}} \tag{27e}$$

$$q(0|k_R) = q(k_R), \quad q_{\text{sc}}(0|k_R) = q_{\text{sc}}(k_R)$$

$$l = 0, \ldots, N_R - 1 \tag{27f}$$

where N_R is the RG prediction horizon and k_R is the time index of the RG discretization step. The first five constraints represent the discretization of the differential-algebraic system (IV-A). The last three constraints contain physical limits of the power converter ($\underline{i}_B, \overline{i}_B$), SoC limits, and the initial values of the battery and SC SoC.

The cost function J_R penalizes battery stress (first term) and power losses (second term)

$$\tilde{p}(l|k_R) = R_{\text{sc}} i_{\text{sc}}^2(l|k_R) + R i^2(l|k_R) + R_B i_B^2(l|k_R). \tag{28}$$

The weight β_R included in the cost function allows the designer to exploit tradeoffs between battery stress and energy losses. After solving (27f), the RG produces a reference trajectory for the SC SoC, $q^*_{\text{sc}}(l|k_R)$, which is tracked by the lower level controller.

B. Low-Level Layer: Tracking and Balancing Controller

The design of the low-level controller relies on the *qTSL* formulation (presented in Section III) extended with one additional cost term. This additional cost penalizes tracking errors, i.e., differences between the SC SoC q_{sc} and the reference q^*_{sc}. The resulting MPC-based tracking and balancing controller is formulated as

$$\min_{\mathbf{u}(l|k)} \quad J_{qTSL} + J_{\text{sc}} \quad \text{s.t.} \quad (22\text{b}) - (22\text{i})$$

where $J_{\text{sc}} = \alpha_{\text{sc}} \sum_{l=1}^{N} [q_{\text{sc}}(l|k) - q^*_{\text{sc}}(l|k)]^2$ embodies the summation of quadratic tracking errors weighted by α_{sc}. Note that, since the RG sample time is usually higher than the low-level controller, $q^*_{\text{sc}}(l|k_R)$ needs be interpolated to generate $q^*_{\text{sc}}(l|k)$ at a higher sample rate. A linear interpolation was employed in this work.

C. Calibration Guidelines and Analysis

The ML $qSTL$ depends on a larger number of weights and tuning parameters, including (N_R, β_R, and τ_s^R) for the RG and ($N, \alpha_q, \alpha_T, \alpha_{\text{SL}}, \beta, \alpha_{\text{sc}}$, and τ_s) for tracking and balancing. To calibrate these parameters, we employed the following guidelines.

1) Prediction Horizons: The RG prediction horizon N_R should be large enough to anticipate and prepare the batteries and SCs for future loads. On the other hand, the prediction horizon for the balancing and tracking layer (N) should be small to reduce computational time (which is significantly higher in the tracking and balancing layer due to the complex *module-level* prediction model). From the analysis present in Fig. 7, N_R equivalent to 200 s provides a good level of stress

reduction in the battery, while a prediction of 10 s is sufficient for balancing tasks.

2) Sample Times: The low-level tracking layer operates with $\tau_s = 1$ s, which is line with the typical sample time of vehicular driving cycles; the RG layer was defined with $\tau_s^R = 5$ s, which provided a good compromise between computational load and performance.

3) Weights for Balancing, Stress and Losses: The weights ($\alpha_q, \alpha_T, \alpha_{\text{SL}}$, and β) can be parameterized with the values obtained during the calibration of the SL MPC. Since both the RG and tracking layers have the same tradeoff goals for battery stress and energy losses, it is desirable to set $\beta_R = \beta$.

4) Weight for SoC Tracking: α_{sc} allows us to weight how closely the reference q_{sc}^* is tracked. For example, Fig. 13(a) shows the results of q_{sc}^* tracking for different weights α_{sc} under the load conditions described in Section III-B. Using a small weight ($\alpha_{\text{sc}} = 1$) causes a quick drop of the SC SoC; the RG increases the reference q_{sc}^* over short bursts as a means to raise q_{sc}. However, since α_{sc} is too small, the SoC tracking is ineffective. Using a large weight ($\alpha_{\text{sc}} = 10^7$) produces a much tighter tracking of q_{sc}^*; interestingly, it also yields results closer to the SL tuned with long preview horizon ($N\tau_s = 200$ s).

The tight tracking of the reference q_{sc}^* brings several additional benefits. As shown in Fig. 13(b), the ML MPC with higher α_{sc} decreases the stress factors (current and temperature) while keeping the SoC and temperature balancing costs low. Overall, the ML MPC with high α_{sc} performs similar to the SL in most of the metrics, but at the expense of higher energy losses.

The ML MPC also reduced the computational effort necessary to solve the *qTSL* problem. In particular, the average calculation times obtained when running the SL MPC ($N\tau_s = 200$ s) algorithm on the HBS embedded control platform were 18.24 s. The ML ($N_R\tau_s^R = 200$ s and $N\tau_s = 10$ s) requires less than 0.51 s, divided into 0.21 s for solving the reference generation problem, and 0.30 for the low-level tracking controller. Thus, the ML variant allows us to obtain similar performance levels as the SL, but with a fraction of the computational times, paving the way for its real-time implementation.

V. Experimental Validation

The *qTSL* control framework described in the previous sections was validated in a small-scale HBS laboratory prototype (Figs. 14 and 15).

A. Description of the HBS Prototype

The prototype is composed of a small battery pack with 12 lithium NMC cells (LG 18650HG2). The cells were arranged in a 4P3S configuration. Each module has four cells connected in parallel, while the pack is composed of three modules connected in series (Fig. 15). We further considered that modules #1 and #2 have one faulty cell each (disconnected from the battery pack), reducing the number of usable cells in these modules to three. All cells in module #3 are healthy. As a result of these faults, the battery pack has significant imbalances that need to be managed by the controller, e.g., modules #1 and #2 have 75% of the capacity of module #3.

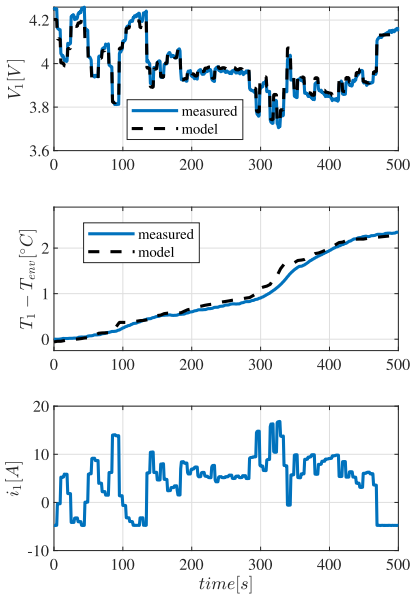


Fig. 16. Validation of the MPC prediction model for battery module #1 (modules #2 and #3 provide similar fitting results and are omitted due to space constraints).

The SC pack is composed of six cells based on Maxwell BCAP0310 P270 T10 and arranged in a 2S3P configuration.

The balancing circuits are based on dual half-bridge power converters [41], providing galvanic isolation and bi-directional power flow capabilities. Each converter has a proportional + integral current loop, implemented on a TI TMS320F28377S microcontroller, which tracks the reference currents ($\mathbf{u}$) through manipulation of the converter's power semiconductors [42]. In addition, to command the load power requested to the HBS, the programmable load *BK Precision* 8500 was configured to track a downscaled version of the US06 driving cycle. This driving cycle information is also shared with the *qTSL* controller to enable load preview.

Both SL and ML variants of the *qTSL* framework were implemented in an embedded control platform based on Raspberry pi 3 [40] with a Quad Core 1.2-GHz BCM2837 64 bit CPU and 2 GB of memory. The resulting optimization problems were solved using the Gekko Toolbox [43], a rapid prototyping framework for numerical optimization, and the IPOP numerical solver [39]. To prevent numerical infeasibility, the constraints (22h) and (27e) were relaxed with slack variables and the MPC cost augmented with an additional term that penalizes the ∞-norm (maximum) constraint violation [44].

B. Model Validation and State Estimation

The *qTSL* controller requires access to three types of states: 1) temperature of the battery modules (T_j); 2) SoC of the battery modules (q_j); and 3) SoC of the SC (q_{sc}). The first is measured by negative temperature-coefficient (NTC) thermistors (NTCLE300E3103SB) that are fit to the exterior of the battery cells. To simplify implementation, a single NTC module is employed for each battery module. The SoC of the battery modules is obtained through the Kalman filter techniques [45]. This combines voltage and current measurements (v_j, i_j) together with an equivalent electric circuit model of the battery modules to produce the SoC estimation. This model is

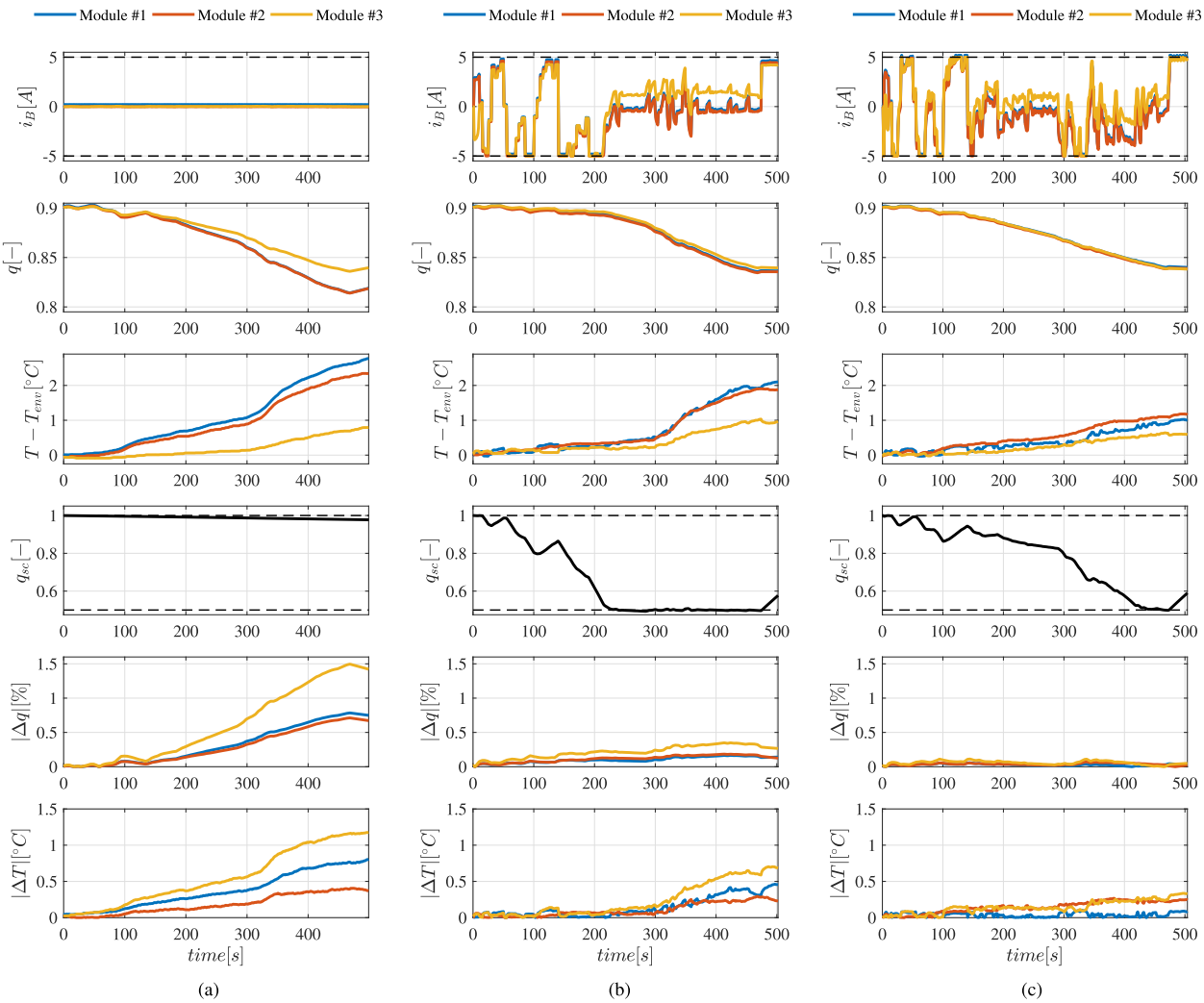


Fig. 17. Experimental results of different control variants. Note: dashed lines indicate operational limits. (a) No control. (b) SL. (c) ML.

based on the equivalent circuit discussed in Section II-B and Fig. 2, augmented with a resistance–capacitor pair connected in series to improve estimation accuracy. Finally, the SoC of the SCs is computed based on SC voltage and current measurements and inversion of (5), i.e., $q_{\text{sc}} = (v_{\text{sc}} + R_{\text{sc}} i_{\text{sc}})/k_{\text{sc}}$.

The parameters of the *qTSL* prediction model, summarized in Table II, were obtained from a combination of different sources. Components' datasheets provided the nominal values for the SC parameters ($Q_{\text{sc}}, C_{\text{sc}}, R_{\text{sc}}$, and k_{sc}) and nominal battery charge capacity (Q_j), while the specific heat capacity data from [46] and the battery mass allowed us to compute the thermal capacity $C_{h,j}$. Due to the larger intercell separation present in our experimental setup (see Fig. 14), we assumed that the conductive thermal resistance $R_{\text{cnd},j}$ is very large and the heat flow between neighboring modules is negligible ($Q_{c,j} \approx 0$). Calibration tests were employed for estimating the remaining parameters. In particular, open-circuit voltage at different SoC points (q_j and $v_{\text{oc},j}$) was collected and then fit with an affine voltage curve ($v_{\text{oc},j}(q_j) = a_j + b_j q_j$) through least square methods. The parameters (R_j and $R_{\text{cov},j}$) were also identified through least square methods, designed to minimize the error between measured and predicted voltage/temperature. Fig. 16 shows the overall fitting results of the prediction model for battery module #1 (the remaining modules have a similar prediction accuracy). Both the predicted voltage and temperature closely follow the experimental measurements, yielding small average fitting errors: 17 mV for the voltage and 0.11 °C for temperature.

C. Results and Analysis

The following control variants were experimentally evaluated on the HBS prototype.

1) *No Control* ($\mathbf{u} = 0$), a battery-only solution that does not make use of the balancing hardware or SCs.
2) *SL*: Single-layer MPC with a short (and real-time capable) preview horizon, $N\tau_s = 10$ s.
3) *ML*: Multilayer MPC with prediction horizons $N_R \tau_s^R = 200$ s, $N\tau_s = 10$ s, and emphasis on SoC balancing.

All control parameters employed in the tests are summarized in Table V. The embedded control platform required 0.3 s on average (0.81 s maximum) to solve the low-level MPC problem, while the high-level MPC problem needed 0.21 s on average (1.19 s maximum). These computational times are in compliance with the timing constraints associated with the MPC sampling times (τ_s and τ_s^R).

Fig. 17(a) shows the HBS operation without control. Due to their reduced capacity, the faulty modules (#1 and #2) discharge faster than the healthy module (#3), inducing SoC

TABLE V
CONTROL PARAMETERS EMPLOYED IN THE EXPERIMENTAL TESTS

Variable	Symbol	Single Layer	Multi Layer
Pred. horizon	N	10	10
Sample time	τ_s	1s	1s
SoC bal. weight	α_q	6.84×10^3	6.84×10^3
Temp. bal. weight	α_T	10^{-3}	10^2
Stress/Losses weight	α_{SL}	1	1
Trade-off factor	$\beta = \beta_R$	0.96	0.96
q^*_{sc} weight	α_{sc}	-	3.79×10^6
Pred. horizon (RG)	N_R	-	50
Sample time (RG)	τ_s^R	-	5s

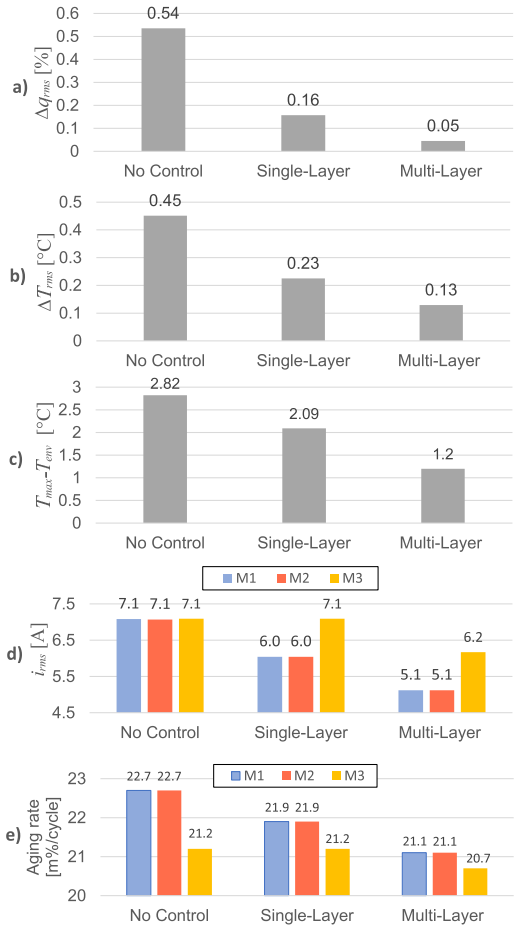


Fig. 18. Summary of performance metrics obtained during the experimental tests for *No Control*, SL and ML MPC controllers. (a) SoC variations. (b) Temperature variations. (c) Temperature increase. (d) Current stress. (e) Average aging rate.

variations (Δq) of up to 1.5%. Cells in the faulty modules are also overloaded with higher currents. Note that the current in the faulty modules is divided among three cells, while healthy module has four cells. This leads to higher temperature increases (2.8 °C in the faulty modules versus 0.8 °C in the healthy module) and temperature imbalances of up to 1.2 °C. It also increases the aging rate of the faulty modules by 6.6% when compared to the healthy module [see Fig. 18(e)].

The results obtained with SL MPC are shown in Fig. 17(b). One can observe a fast discharge of the SC SoC (q_{sc}), reaching the lower SoC limit in less than 220 s. This occurs because of the SL's small preview horizon, which is necessary to ensure real-time execution. Closer inspection of the results also reveals the emergence of two operating modes. In the first, which occurs from 0 to 220 s, the SCs support all battery modules: $i_{B,1}, i_{B,2}$, and $i_{B,3}$ are negative in a large portion of time, which indicates charging of the battery modules. In the second mode, which occurs after 220 s, the controller focuses mainly on SoC and temperature balancing; the balancing circuit discharges the healthy module ($i_{B,3} > 0$) and charges the faulty modules ($i_{B,1}, i_{B,2} < 0$) most of the time to compensate for charge imbalances. From a performance perspective, the SL MPC improves every metric when compared to *No Control*. As shown in Fig. 18, the SL MPC decreases SoC imbalances by 70% and temperature imbalances by 50%; the current stress and aging rate in faulty modules (#1 and #2) is also decreased by 15% and 3.5%, respectively.

Fig. 17(c) shows the ML MPC. There are a few points worth highlighting. First, due to the RG's longer preview horizon, most of the SCs energy is deployed during the later part of the driving cycle (300–400 s) when larger power loads are applied. These results are in line with those observed from simulations in Section IV. Second, the ML MPC provides localized support to the weaker battery; for example, during discharging, faulty modules (#1 and #2) receive more current/support than the healthy module (#3), which helps balance the battery pack. Third, the ML MPC recharges the SCs during regenerative braking periods, e.g., q_{sc} during 100–130 s in Fig. 17(c). This policy allows the SCs to support the faulty modules more effectively during peak load demands that occur later in the driving cycle.

As shown in Fig. 18, these features contribute to a superior performance of the ML. In comparison with *No Control*, the ML reduces SoC imbalances by 91% and temperature imbalances by 70%. In addition, a 57% reduction in the maximum temperature increase (above ambient temperature) is observed, which is due to the reduced current stress (-28%) and aging rate (-7%) in the faulty modules #1 and #2. The ML also decreased the current stress in the healthier module (-12%) in comparison to *No Control* due to better utilization of the SCs energy.

VI. CONCLUSION AND OUTLOOK

A predictive control framework for HBSs was proposed in this work. The control framework was able to optimize multiple objectives, including charge, temperature, stress, and losses (*qTSL*) while enforcing state and current constraints and exploiting load forecast information. Computational and tuning efforts were the main challenges that emerged in the implementation of this framework. To address the first challenge, an ML control architecture was developed. This decomposed the control problem into a high-level layer, responsible for generating SC charge references considering long load forecasts and simple pack-level models, and a low-level layer that tracked the SC charge references and equalized the battery pack using more complex module-level models and short prediction horizons. Due to the exploitation of different model complexities and time scales, the ML architecture was shown to reduce computation effort while still achieving strong overall control performance. To address the second challenge, practical calibration guidelines were proposed. These relied on a sequential process, which helped the designer to quickly

tune the weights of the predictive controllers. Experimental validation with a small-scale laboratory prototype demonstrated the ability of the ML framework to equalize charge and temperature in a battery pack with significant capacity imbalances. The control framework also exploited the inclusion of SCs in the balancing system to reduce battery stress (current) 28%, aging rate 7%, and temperature 57% when compared to *No Control* operation.

While this article focuses on implementation to explore the potential performance benefits of ML MPC for HBSs, theoretical techniques have been proposed in the literature for ensuring stability and/or robustness of hierarchical and distributed control frameworks. For example, techniques developed for energy systems include passivity analysis [47], contraction theory [48], and constraint tightening [26]. Future work will draw on this prior literature to provide analytical guarantees for the proposed *qTSL* framework. We will also investigate large-scale applications of this framework and improve the computational efficiency of the tracking control layer, e.g., by exploiting suboptimal MPC approaches [49] or reference governors [50] with reduced computational footprint.

APPENDIX

A. Parameters of the Aging Model

$\theta_1 = 2.047 \times 10^{-4}, \theta_2 = 8.774 \times 10^{-3}, \theta_3 = 1.053, \theta_4 = 9.627, \theta_5 = 4.145, \theta_6 = 5.851 \times 10^2, \theta_7 = 2.473 \times 10^{-2}, \theta_8 = 8.568 \times 10^{-3}, \theta_9 = -3.29 \times 10^2,$ and $\theta_{10} = 1.966 \times 10^1$.

REFERENCES

[1] N. Lutsey, "Modernizing vehicle regulations for electrification," Int. Council Clean Transp., San Francisco, CA, USA, Tech. Rep., 2018.

[2] S. Koohi-Fayegh and M. A. Rosen, "A review of energy storage types, applications and recent developments," *J. Energy Storage*, vol. 27, Feb. 2020, Art. no. 101047.

[3] B. Kenney, K. Darcovich, D. D. MacNeil, and I. J. Davidson, "Modelling the impact of variations in electrode manufacturing on lithium-ion battery modules," *J. Power Sources*, vol. 213, pp. 391–401, Sep. 2012.

[4] M. Jafari, K. Khan, and L. Gauchia, "Deterministic models of Li-ion battery aging: It is a matter of scale," *J. Energy Storage*, vol. 20, pp. 67–77, Dec. 2018.

[5] M.-K. Tran and M. Fowler, "A review of lithium-ion battery fault diagnostic algorithms: Current progress and future challenges," *Algorithms*, vol. 13, no. 3, p. 62, Mar. 2020.

[6] J. V. Barreras, D. Frost, and D. Howey, "Smart balancing systems: An ultimate solution to the weakest cell problem?" *IEEE Veh. Technol. Soc. Newslett.*, Mar. 2018.

[7] A. Kelkar, Y. Dasari, and S. S. Williamson, "A comprehensive review of power electronics enabled active battery cell balancing for smart energy management," in *Proc. IEEE Int. Conf. Power Electron., Smart Grid Renew. Energy (PESGRE)*, Jan. 2020, pp. 1–6.

[8] J. Carter, Z. Fan, and J. Cao, "Cell equalisation circuits: A review," *J. Power Sources*, vol. 448, Feb. 2020, Art. no. 227489.

[9] H. Rahimi-Eichi, U. Ojha, F. Baronti, and M.-Y. Chow, "Battery management system: An overview of its application in the smart grid and electric vehicles," *IEEE Ind. Electron. Mag.*, vol. 7, no. 2, pp. 4–16, Jun. 2013.

[10] L. McCurlie, M. Preindl, and A. Emadi, "Fast model predictive control for redistributive lithium-ion battery balancing," *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 1350–1357, Feb. 2017.

[11] W. Han, T. Wik, A. Kersten, G. Dong, and C. Zou, "Next-Generation battery management systems: Dynamic reconfiguration," *IEEE Ind. Electron. Mag.*, vol. 14, no. 4, pp. 20–31, Dec. 2020.

[12] J. V. Barreras, C. Pinto, R. de Castro, E. Schaltz, S. J. Andreasen, and R. E. Araujo, "Multi-objective control of balancing systems for li-ion battery packs: A paradigm shift?" in *Proc. IEEE Vehicle Power Propuls. Conf. (VPPC)*, Oct. 2014, pp. 1–7.

[13] W. Wang and M. Preindl, "Dual cell links for battery-balancing auxiliary power modules: A cost-effective increase of accessible pack capacity," *IEEE Trans. Ind. Appl.*, vol. 56, no. 2, pp. 1752–1765, Mar. 2020.

[14] A. Singer, F. Helling, T. Weyh, J. Jungbauer, and H. Pfisterer, "Modular multilevel parallel converter based split battery system (M2B) for stationary storage applications," in *Proc. 19th Eur. Conf. Power Electron. Appl.*, Sep. 2017, p. 1.

[15] M. Kuder, J. Schneider, A. Kersten, T. Thiringer, R. Eckerle, and T. Weyh, "Battery modular multilevel management (BM3) converter applied at battery cell level for electric vehicles and energy storages," in *Proc. PCIM Int. Exhib.*, 2020, pp. 1–5.

[16] R. de Castro, C. Pinto, J. Varela Barreras, R. E. Araújo, and D. A. Howey, "Smart and hybrid balancing system: Design, modeling, and experimental demonstration," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 11449–11461, Dec. 2019.

[17] E. Chemali, M. Preindl, P. Malysz, and A. Emadi, "Electrochemical and electrostatic energy storage and management systems for electric drive vehicles: State-of-the-art review and future trends," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 4, no. 3, pp. 1117–1134, Sep. 2016.

[18] M. Preindl, "A battery balancing auxiliary power module with predictive control for electrified transportation," *IEEE Trans. Ind. Electron.*, vol. 65, no. 8, pp. 6552–6559, Aug. 2018.

[19] A. Pozzi, M. Zambelli, A. Ferrara, and D. M. Raimondo, "Balancing-aware charging strategy for series-connected lithium-ion cells: A nonlinear model predictive control approach," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 5, pp. 1862–1877, Sep. 2020.

[20] F. Altaf, B. Egardt, and L. J. Mårdh, "Load management of modular battery using model predictive control: Thermal and state-of-charge balancing," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 1, pp. 47–62, Jan. 2017.

[21] D. J. Docimo and H. K. Fathy, "Analysis and control of charge and temperature imbalance within a lithium-ion battery pack," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 4, pp. 1622–1635, Jul. 2019.

[22] A. Pozzi, M. Torchio, R. D. Braatz, and D. M. Raimondo, "Optimal charging of an electric vehicle battery pack: A real-time sensitivity-based model predictive control approach," *J. Power Sources*, vol. 461, Jun. 2020, Art. no. 228133.

[23] A. Allam and S. Onori, "Exploring the dependence of cell aging dynamics on thermal gradient in battery modules: A PDE-based time scale separation approach," in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 2380–2385.

[24] T. Gewald, A. Candussio, L. Wildfeuer, D. Lehmkuhl, A. Hahn, and M. Lienkamp, "Accelerated aging characterization of lithium-ion cells: Using sensitivity analysis to identify the stress factors relevant to cyclic aging," *Batteries*, vol. 6, no. 1, p. 6, Jan. 2020.

[25] R. R. Negenborn and J. M. Maestre, "Distributed model predictive control: An overview and roadmap of future research opportunities," *IEEE Control Syst.*, vol. 34, no. 4, pp. 87–97, Aug. 2014.

[26] M. R. Amini, I. Kolmanovsky, and J. Sun, "Hierarchical MPC for robust eco-cooling of connected and automated vehicles and its application to electric vehicle battery thermal management," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 1 pp. 316–328, Jan. 2020.

[27] J. P. Koeln, H. C. Pangborn, M. A. Williams, M. L. Kawamura, and A. G. Alleyne, "Hierarchical control of aircraft electro-thermal systems," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 4, pp. 1218–1232, Jul. 2020.

[28] R. de Castro, H. Pereira, R. E. Araujo, J. V. Barreras, and R. H. Pangborn, "Multi-layer control for hybrid balancing systems," in *Proc. 5th IEEE Conf. Control Technol. Appl.*, 2021.

[29] K. Liu, Z. Yang, X. Tang, and W. Cao, "Automotive battery equalizers based on joint switched-capacitor and buck-boost converters," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12716–12724, Nov. 2020.

[30] M. Evzelman, M. M. Ur Rehman, K. Hathaway, R. Zane, D. Costinett, and D. Maksimovic, "Active balancing system for electric vehicles with incorporated low-voltage bus," *IEEE Trans. Power Electron.*, vol. 31, no. 11, pp. 7887–7895, Nov. 2016.

[31] W. Waag, C. Fleischer, and D. U. Sauer, "Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles," *J. Power Sources*, vol. 258, pp. 321–339, Jul. 2014.

[32] X. Lin, H. E. Perez, J. B. Siegel, A. G. Stefanopoulou, Y. Ding, and M. P. Castanier, "Parameterization and observability analysis of scalable battery clusters for onboard thermal management," *Oil Gas Sci. Technol.*, vol. 68, no. 1, pp. 165–178, 2011.

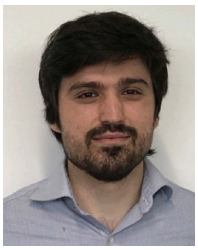
[33] A. Berrueta, A. Ursúa, I. S. Martín, A. Eftekhari, and P. Sanchis, "Supercapacitors: Electrical characteristics, modeling, applications, and future trends," *IEEE Access*, vol. 7, pp. 50869–50896, 2019.

[34] C. R. Birkl, M. R. Roberts, E. McTurk, P. G. Bruce, and D. A. Howey, "Degradation diagnostics for lithium ion cells," *J. Power Sources*, vol. 341, pp. 373–386, Feb. 2017.
[35] S. Saxena, Y. Xing, D. Kwon, and M. Pecht, "Accelerated degradation model for C-rate loading of lithium-ion batteries," *Int. J. Elect. Power Energy Syst.*, vol. 107, pp. 438–445, May 2019.
[36] A. Maheshwari, "Modelling, aging and optimal operation of lithium-ion batteries," Ph.D. dissertation, Dept. Elect. Comput. Eng., Techn. Univ. Eindhoven, Eindhoven, The Netherlands, 2018.
[37] A. Sciarretta and A. Vahidi, *Energy-Efficient Driving of Road Vehicles*. Cham, Switzerland: Springer, 2020.
[38] T. J. Barlow, S. Latham, I. S. McCrae, and P. G. Boulter, "A reference book of driving cycles for use in the measurement of road vehicle emissions," TRL, Tech. Rep. PPR354, 2009.
[39] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, May 2006.
[40] *Raspberry Pi 4 Technical Specifications*. Accessed: Sep. 25, 2020. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/
[41] H. Li and F. Z. Peng, "Modeling of a new ZVS bi-directional dc-dc converter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 1, pp. 272–283, Jan. 2004.
[42] C. Pinto, "Sizing and energy management of a distributed hybrid energy storage system for electric vehicles," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Porto, Porto, Portugal, 2018.
[43] L. Beal, D. Hill, R. Martin, and J. Hedengren, "GEKKO optimization suite," *Processes*, vol. 6, no. 8, p. 106, Jul. 2018.
[44] J. M. Maciejowski, *Predictive Control With Constraint*. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.
[45] G. L. Plett, *Battery Management Systems, volume II: Equivalent-Circuit Methods*. Norwood, MA, USA: Artech House, Dec. 2015.
[46] J. V. Barreras *et al.*, "An advanced HIL simulation battery model for battery management system testing," *IEEE Trans. Ind. Appl.*, vol. 52, no. 6, pp. 5086–5099, Nov. 2016.
[47] H. C. Pangborn, J. P. Koeln, and A. G. Alleyne, "Passivity and decentralized MPC of switched graph-based power flow systems," in *Proc. Annu. Amer. Control Conf.*, 2018, pp. 198–203.
[48] Y. Long, S. Liu, L. Xie, and K. H. Johansson, "Distributed nonlinear model predictive control based on contraction theory," *Int. J. Robust Nonlinear Control*, vol. 28, no. 2, pp. 492–503, Jan. 2018.
[49] D. Liao-McPherson, M. M. Nicotra, and I. Kolmanovsky, "Time-distributed optimization for real-time model predictive control: Stability, robustness, and constraint satisfaction," *Automatica*, vol. 117, Jul. 2020, Art. no. 108973.
[50] E. Garone, S. Di Cairano, and I. V. Kolmanovsky, "Reference and command governors for systems with constraints: A survey on theory and applications," *Automatica*, vol. 75, pp. 306–328, Jan. 2017.

Ricardo de Castro (Senior Member, IEEE) received the Licenciatura and Ph.D. degrees in electrical and computer engineering from the Faculty of Engineering, University of Porto, Porto, Portugal, in 2006 and 2013, respectively.

From 2007 to 2008, he was an Entrepreneur with the WeMoveU Project, targeting the development of powertrain control solutions for lightweight electric vehicles. From 2013 to 2020, he was with German Aerospace Center (DLR), Institute of System Dynamics and Control (SR), Wessling, Germany, where he developed enabling technologies for electric mobility and automated driving. In 2021, he joined the University of California at Merced, Merced, CA, USA, as an Assistant Professor.

Dr. de Castro has been an Expert Evaluator for the European Union. He was the Chair of Technical Tracks on Vehicular Electronics and Intelligent Transportation Systems and IEEE Vehicle Power and Propulsion Conference (VPPC) from 2017 to 2018. He has been an Editor of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, an Associate Editor of IEEE ACCESS, and a Guest Editor of *Energies* journal.



Hélder Pereira (Graduate Student Member, IEEE) was born in Porto, Portugal. He received the M.S. degree in electric engineering from the University of Porto, Porto, in 2019.

He is currently a Software Engineer with Continental Engineering Services, Porto. His research interests include embedded systems, predictive control, and electric vehicles.

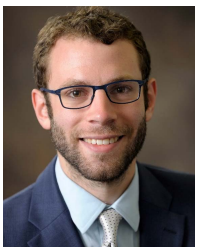
Rui Esteves Araújo (Member, IEEE) received the Diploma degree in electrical engineering (five-year university degree) and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Porto, Porto, Portugal, in 1987, 1992, and 2001, respectively.

From 1987 to 1988, he was an Electrotechnical Engineer with the Project Department of Adira, Metal Forming Solutions Company, Porto, and from 1988 to 1989, he was a Researcher with INESC, Porto. Since 1989, he has been with the University of Porto, where he is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Faculty of Engineering. He is a Senior Researcher with INESC TEC, Porto, focusing on control theory and its industrial applications to motion control, electric vehicles, and renewable energies.

Jorge Varela Barreras (Member, IEEE) received the M.Sc. degree in power electronics (two-year degree) from Aalborg University, Aalborg, Denmark, in 2009, the Industrial Engineering degree with specialization in electrical technology (six-year degree) from the University of Vigo, Vigo, Spain, in 2010, and the Ph.D. degree from Aalborg University, in 2017.

He combines an academic role as a Battery Researcher with Imperial College London, London, U.K., with an industrial role as an EV Expert at CTAG, Pontevedra, Spain. He held a post-doctoral position with the University of Oxford, Oxford, U.K., and a co-founder of a consulting company. He was involved in 16 international research projects with a combined budget over 75M euros. He has coauthored 31 peer-reviewed articles, including 11 journal articles and two book chapters. His interests include EVs, energy storage systems, batteries, and battery management.

Dr. Barreras is involved in the community as the Joint Chairman of IEEE U.K. and Ireland Education Society, and Editorial Board Member of *eTransportation* (Elsevier) and World Electric Vehicle Journal (WEVJ) (MDPI). He was the Co-Chairman of the Oxford Research Staff Society and the Founder Member of the Danish Battery Society.



Herschel C. Pangborn (Member, IEEE) received the B.S. degree in mechanical engineering from The Pennsylvania State University, University Park, PA, USA, in 2013, and the M.S. and Ph.D. degrees in mechanical engineering from the University of Illinois at Urbana–Champaign, Urbana, IL, USA, in 2015 and 2019, respectively.

He was the National Science Foundation (NSF) Graduate Research Fellow and a Summer Faculty Fellow with U.S. Air Force Research Laboratory, Wright-Patterson AFB, OH, USA. He is an Assistant Professor with the Department of Mechanical Engineering, The Pennsylvania State University. His current research interests include dynamic modeling, design, and control of electrothermal systems, including model predictive control and hierarchical control.

Dr. Pangborn was a recipient of the Best Paper Award from the Energy Systems Technical Committee of the ASME Dynamic Systems and Control Division.