Active Cell Balancing by Model Predictive Control for Real Time Electric Vehicles Range Extension

Jun Chen, *Senior Member, IEEE*, Aman Behal, *Senior Member, IEEE* and Chong Li, *Senior Member, IEEE*

***Abstract*—This paper studies the active cell balancing problem by using model predictive control (MPC) for real time range extension. Specifically, three MPC formulations are proposed and compared. In the first formulation, MPC is set up to be a tracking controller to force all cells to follow the same trajectory generated by a nominal cell model. In the second formulation, MPC is set to maximize the lowest cell SOC/voltage to extend the operating range of battery pack, while in the last formulation, MPC is set to minimize the difference between the highest and lowest cell SOC/voltages. All three MPC controllers are evaluated using both steady state and transient conditions to assess their effectiveness, and a range extension of 9% is found for dynamic driving cycle and 7% for steady state condition. Furthermore, it is shown that the first formulation is favored for transient conditions while the last two formulations are comparably favored for steady state conditions. Comparing to the existing work, our approaches achieve similar range extension, without making the assumption that the final battery state-of-charge is known in advance, which can be a very restrictive assumption given the current technology. Therefore, our approaches are more applicable. Finally, real time implementability is demonstrated via throughput analysis.**

***Index Terms*—Battery balancing control, equivalent circuit model, model predictive control, quadratic programming**

I. Introduction

Recently, electric vehicles (EV) and stationary smart grid applications have been considered as among the most promising ways to combat the environment challenge, creating a high momentum for efficient battery systems [1]. Among many other choices, Lithium-Ion (Li-Ion) battery cells are dominating EV applications due to their high power and energy density. Due to manufacturing variation and/or different aging conditions, Li-Ion battery cells can experience state-of-charge (SOC) and voltage imbalance [2]. Such imbalance inevitably degrades the battery performance, reduces EV range, and leads to safety issues such as overheating and thermal runaway [3]–[6]. The battery cell imbalance issue can be mitigated using a cell balancing circuit, such as flyback DC/DC converter [6], [7] and half-bridge converter [8]. This is called cell balancing in literature, and can be either *dissipative* or *nondissipative*. Dissipative method removes charges from higher SOC cell without reusing them [9], while nondissipative method moves charges from cells to cells, [6], [10]. In this paper, we focus on nondissipative cell balancing control, which is also called active cell balancing and has less energy waste.

The goal of active cell balancing is to push all cell's voltages away from the lower bound, below which a cell would fail and lead to the failure of the entire battery pack. To achieve this goal, several control techniques have been studied in the literature [1], [11]–[13]. For example, [12] proposed a rule based balancing control algorithm for groupwise balancing and demonstrate the robustness and performance through realistic driving profile. Authors in [1] designed a power converter circuit that allows many to many balancing, and developed a fast simulation model for analysis. Heuristic strategy was developed and shows promising results in [1], [13].

Though these works demonstrate promising results, they rely on simple control methodology and do not fully unlock the potential of active cell balancing. To address this issue, advanced control methods, such as model predictive control (MPC), have been investigated in the literature for active cell balancing [4], [10], [14]–[17]. In [10], an auxiliary power module was designed to perform balancing during charging, and two MPC strategies were considered using linear SOC model, one optimizing charging and balancing simultaneously while the other using separate controls for charging and balancing. Linear MPC was also adopted in [4] to consider different balancing objectives, namely, SOC, voltage, and charges, each configuration aiming at tracking a reference trajectory generated by assuming the final SOC was known in advance. Though such an assumption was very restrictive, a 5% range increase was shown through simulation. Reference [3] cast the balancing control problem as a reachability analysis problem, and showed the benefit on range extension on a short driving cycle. Reference [15] utilized nonlinear MPC to simultaneously minimize SOC imbalance and energy waste due to balancing current. Similarly, reference [14] also considered energy waste in the cost function for minimization. To fit into a microcontroller, [16] considered fast MPC where the objective is to minimize the time to balance with a linear dynamic model for balancing current, resulting in a linear programming problem that is suitable for an embedded environment. Finally, [17] studies the hierarchical MPC control for cell balancing, where the higher level control optimizes modules balancing current and the lower level control optimizes the balancing currents within each module.

However, there are several limitations with the aforementioned MPC-based active cell balancing. For example, only tracking controller was considered in [4], [15], [17], which *in-*

This work was supported in part by the faculty startup fund from Oakland University and Michigan Space Grant Consortium through Grant 80NSSC20M0124. (*Jun Chen is the corresponding author.*)

Jun Chen is with the Department of Electrical and Computer Engineering, Oakland University, Rochester, MI 48309, USA (email: junchen@oakland.edu).

Aman Behal is with the Electrical and Computer Engineering and NanoScience Technology Center, University of Central Florida, Orlando, FL 32816, USA (e-mail: abehal@ucf.edu).

Chong Li is with the Department of Electrical Engineering, Columbia University, NYC, 10027, USA (email: cl3607@columbia.edu).

directly implements the balancing goal. Moreover, the assumption in [4] that the final battery SOC is known in advance for reference generation, which can be very restrictive given the current technology and make the developed control algorithm not practical for short-term deployment. Furthermore, most of the work either utilize linear prediction model [4], [10], [14] that can result in over simplification or nonlinear model [15] that cause high computation cost, and a better approach that intermingle these two, e.g., linear parametric varying (LPV) prediction model, is needed.

To address these limitations, In this paper, we study MPC for active cell balancing problem for EV range extension, and investigate three balancing objectives for MPC. In the first objective, MPC is set to track the SOC/voltages of all cells to follow a short term reference that is generated by using a nominal battery cell model. This setup is similar to the MPC formulation of [4], without assuming that the final SOC at the end of the drive cycle is known in advance. In the second objective, MPC is set to maximize the lowest cell SOC/voltage, as opposed to track all cells' SOC/voltages. In the last objective, MPC is set to minimize the difference between the highest and the lowest cell SOC/voltages. Note that the three proposed MPC formulations realize active balancing control by using different cost functions, whose effectiveness will be investigated. Numerical simulation of these three MPCs are presented and it is found that the first formulation is favored for transient conditions while the last two formulations are comparably favored for steady state conditions. A 9% range extension is shown for dynamic driving cycle (e.g., Federal Test Procedure [FTP]) and 7% for steady state condition. Finally, real time implementability is demonstrated by throughput analysis.

Comparing to the existing work, our approaches achieve 7% range extension and is higher than the 5% reported in [4], without assuming that the final battery state-of-charge is known in advance, which can be a very restrictive assumption given the current technology. Furthermore, the proposed MPC formulations allow direct maximization of the lowest SOC/voltage, instead of indirectly achieving this through tracking controller. Lastly, the developed MPC can be cast into linearly constrained quadratic programming problem, which has been proven to be suitable for real-time implementation. A preliminary version of this work [18] has been accepted for presentation and publication in *2021 IEEE Conference on Decision and Control*. This submitted manuscript extends the conference version by completing literature review, including details on control design, and improving the simulation studies and supplementing with additional results and analysis.

The rest of this paper is organized as follows. Section II presents the equivalent circuit model for each cell and the whole battery pack, while Section III formulates the optimal control problems and three MPC setups. Section IV presents numerical results, and the paper is concluded in Section V.

II. Battery Model

The series connected battery considered here is shown in Fig. 1, where N cells are stacked to provide the requested current i to the load, e.g. an EV in this case. The cell balancer, in this case a power converter circuit, can then be used to draw current u^n from an arbitrary cell n and transport it to another cell j, to reduce the variations of SOC and voltage among the series connected cells.

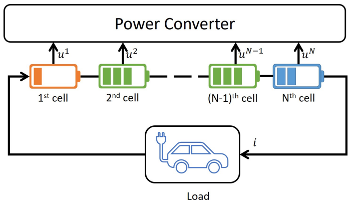

Fig. 1. Structure of series connected battery cell with balancing current.

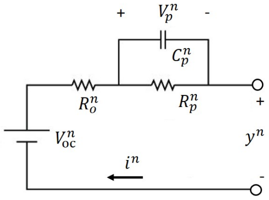

Fig. 2. Equivalent circuit model of a battery cell.

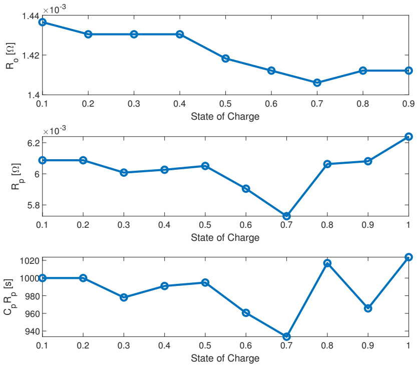

Fig. 3. Parameters for a nominal cell, adopted from [22].

A. Equivalent Circuit Model

The dynamics of each battery cell can be modeled using an equivalent circuit model (ECM), which provides a good balance between good accuracy and low computational cost and has been widely used in the literature [4], [19]–[25]. We briefly describe ECM as follows. More details can be found in aforementioned references.

The ECM used to model a battery cell is shown in Fig. 2, where the superscript n denotes the nth cell, V_{oc}^n is the open circuit voltage, y^n is the terminal voltage, R_o^n, R_p^n, and C_p^n are resistance and capacitor of the ECM, respectively, and i^n is the battery pack current. We use the convention that positive value of i^n indicates discharging from the battery and negative indicates charging to the battery. Denote s^n as the remaining SOC of cell n. The dynamics are then specified by

$$\dot{s^n} = -\eta^n \frac{i^n}{3600 C^n} \tag{1a}$$

$$\dot{V}_p^n = -\frac{V_p^n}{R_p^n C_p^n} + \frac{i^n}{C_p^n} \tag{1b}$$

$$y^n = V_{oc}^n - V_p^n - i^n R_o^n, \tag{1c}$$

where η^n is the coulombic efficiency of cell n, C^n is the cell capacity with unit of Amp Hour, and V_p^n is the relaxation voltage over the RC component. Note that V_{oc}^n, R_o^n, R_p^n, and C_p^n are all dependent of s^n, making (1) a nonlinear model, i.e., nonlinear with respect to states. Fig. 3 depicts an example of such dependency for R_o^n, R_p^n, and C_p^n for a nominal cell. Furthermore, $V_{oc}^n = -1.9123\,(s^n)^2 + 3.6775\,(s^n) + 2.4348.$ Note that due to manufacturing variation and/or different aging conditions, the dependency of V_{oc}^n, R_o^n, R_p^n, and C_p^n on s^n can be different for each cell n, resulting different characteristics for each cell.

Denote $\zeta^n := [s^n, V_p]^T$ where $\cdot^T$ denotes matrix/vector transpose, then one can write (1) as

$$\dot{\zeta}^n = f^n(\zeta^n, i^n), \qquad y^n = g^n(\zeta^n, i^n), \tag{2a}$$

where functions $f^n(\zeta, i^n)$ and $g^n(\zeta, i^n)$ are defined by (1). Considering the battery structure in Fig. 1, the current i^n drawn through cell n equals the pack current i plus balancing current u^n. Therefore, we can rewrite (2) as

$$\dot{\zeta}^n = f^n(\zeta^n, i + u^n), \qquad y^n = g^n(\zeta^n, i + u^n). \tag{3a}$$

Define $\zeta = [\zeta^1, \zeta^2, \ldots, \zeta^N]^T$ as the state vector for the entire battery pack and y as the terminal voltage of the battery pack, then we have

$$\dot{\zeta} = \begin{bmatrix} f^n(\zeta^1, i + u^1) \\ f^n(\zeta^2, i + u^2) \\ \vdots \\ f^n(\zeta^N, i + u^N) \end{bmatrix} \tag{4a}$$

$$y = \sum_{n=1}^{N} y^n = \sum_{n=1}^{N} g^n(\zeta^n, i + u^n). \tag{4b}$$

B. Model Linearization and Discretization

To derive a linearized discrete-time model for MPC, denote for time k the current state estimate and output for cell n as $\hat{\zeta}_k^n = [\hat{s}_k^n; \hat{V}_{p,k}^n]$ and $\hat{y}_k^n$. Denote $\hat{u}_k^n$ and $\hat{i}_k$ as the nominal balancing current and pack current, respectively. Note that one can simply choose the balancing current applied at previous control loop as $\hat{u}_k^n$ and the requested pack current at time k as $\hat{i}_k$. Further denote $\delta\zeta_k^n = [\delta s_k^n; \delta V_{p,k}^n]$, δy_k^n, δi_k and δu_k^n as the deviations from their nominal values. Then, one can linearize (3) to obtain

$$\delta\dot{s}_k^n = -\eta^n \frac{\hat{i}_k + \hat{u}_k^n}{C^n} - \eta^n \frac{1}{C^n}\delta i_k - \eta^n \frac{1}{C^n(\hat{s}_k^n)}\delta u_k^n$$

$$\begin{aligned}
\delta\dot{V}_{p,k}^n =& -\frac{\hat{V}_{p,k}^n}{R_p^n(\hat{s}_k^n)C_p^n(\hat{s}_k^n)} + \frac{\hat{i}_k + \hat{u}_k^n}{C_p^n(\hat{s}_k^n)} \\
&+ \frac{\hat{V}_{p,k}^n}{\left(R_p^n(\hat{s}_k^n)C_p^n(\hat{s}_k^n)\right)^2} \left.\frac{\partial\left(R_p^n(s^n)C_p^n(s^n)\right)}{\partial s^n}\right|_{s^n=\hat{s}_k^n} \delta s_k^n \\
&- \frac{\hat{i}_k + \hat{u}_k^n}{\left(C_p^n(\hat{s}_k^n)\right)^2} \left.\frac{\partial C_p^n(s^n)}{\partial s^n}\right|_{s^n=\hat{s}_k^n} \delta s_k^n \\
&- \frac{1}{R_p^n(\hat{s}_k^n)C_p^n(\hat{s}_k^n)}\delta V_{p,k}^n + \frac{1}{C_p^n(\hat{s}_k^n)}\delta i_k + \frac{1}{C_p^n(\hat{s}_k^n)}\delta u_k^n
\end{aligned}$$

$$\begin{aligned}
y_k^n =& V_{oc}^n(\hat{s}_k^n) - \hat{V}_{p,k}^n - \hat{i}_k R_o^n(\hat{s}_k^n) + \left.\frac{\partial V_{oc}^n(s^n)}{\partial s^n}\right|_{s^n=\hat{s}_k^n} \delta s_k^n \\
&- (\hat{i}_k + \hat{u}_k^n) \left.\frac{\partial R_o^n(s^n)}{\partial s^n}\right|_{s^n=\hat{s}_k^n} \delta s_k^n \\
&- \delta V_{p,k}^n - R_o^n(\hat{s}_k^n)\delta i_k - R_o^n(\hat{s}_k^n)\delta u_k^n.
\end{aligned}$$

Putting everything together, we have

$$\delta\dot{\zeta}_k^n = f^n(\hat{\zeta}_k^n, \hat{i}_k + \hat{u}_k^n) + A_k^c\delta\zeta_k^n + B_k^c\delta u_k^n + B_{d,k}^c\delta i_k \tag{5a}$$

$$y_k^n = g^n(\hat{\zeta}_k^n, \hat{i}_k + \hat{u}_k^n) + C_k^c\delta\zeta_k^n + D_k^c u_k^n + D_{d,k}^c\delta i_k, \tag{5b}$$

where A_k^c is a 2×2 matrix, B_k^c and $B_{d,k}^c$ are 2×1 matrices, C_k^c is a 2×2 matrix, D_k^c and $D_{d,k}^c$ are scalar, with elements specified as follows.

$$\begin{aligned}
A_k^c(1,1) =& 0, \quad A_k^c(1,2) = 0 \\
A_k^c(2,1) =& \frac{\hat{V}_{p,k}^n}{\left(R_p^n(\hat{s}_k^n)C_p^n(\hat{s}_k^n)\right)^2} \left.\frac{\partial\left(R_p^n(s^n)C_p^n(s^n)\right)}{\partial s^n}\right|_{s^n=\hat{s}_k^n} \\
&- \frac{\hat{i}_k + \hat{u}_k^n}{\left(C_p^n(\hat{s}_k^n)\right)^2} \left.\frac{\partial C_p^n(s^n)}{\partial s^n}\right|_{s^n=\hat{s}_k^n} \\
A_k^c(2,2) =& -\frac{1}{R_p^n(\hat{s}_k^n)C_p^n(\hat{s}_k^n)} \\
B_k^c =& B_{d,k}^c = \begin{bmatrix} -\eta^n \frac{1}{C^n(\hat{s}_k^n)} & \frac{1}{C_p^n(\hat{s}_k^n)} \end{bmatrix}^T \\
C_k^c(1,1) =& \left.\frac{\partial V_{oc}^n(s^n)}{\partial s^n}\right|_{s^n=\hat{s}_k^n} - (\hat{i}_k + \hat{u}_k^n) \left.\frac{\partial R_o^n(s^n)}{\partial s^n}\right|_{s^n=\hat{s}_k^n} \\
C_k^c(1,2) =& -1, \quad D_k^c = -R_o^n(\hat{s}_k^n), \quad D_{d,k}^c = D_k^c.
\end{aligned}$$

After discretizing (5) using Euler's forward integration with sampling time T_s, we have

$$\delta\zeta_{k+1}^n = \hat{f}_k^n + A_k\delta\zeta_k^n + B_k\delta u_k^n + B_{d,k}\delta i_k \tag{6a}$$

$$y_k^n = \hat{g}_k^n + C_k\delta\zeta_k^n + D_k u_k^n + D_{d,k}\delta i_k, \tag{6b}$$

with $\hat{f}_k^n = f^n(\hat{\zeta}_k^n, \hat{i}_k + \hat{u}_k^n)T_s$, $A_k = I + A_k^c T_s$, $B_k = B_k^c T_s$, $B_{d,k} = B_k$, $\hat{g}_k^n = g^n(\hat{\zeta}_k^n, \hat{i}_k + \hat{u}_k^n)$, $C_k = C_k^c$, $D_k = D_k^c$, and $D_{d,k} = D_k$.

Note that this linear model is parametric varying, since at each time step k, a new set of matrices are obtained based on the current operating conditions. Furthermore, (6) is scheduled according to SOC s, which is often not measured. However, several techniques in literature that can effectively estimate s according to terminal voltage measurement. See for example [12], [15], [17], [22], [26], [27].

III. Problem Formulation and MPC Setup

A. Optimal Control Problem

Since voltage limit is the most important factor that impacts the battery operational range, the primary objective here is to actively move charge from cell to cell so that all cells stay away from the lowest voltage bound, denoted as $\underline{y}$. Note that if any cell falls below this minimum bound, $\underline{y}$, the cell would fail and lead to the failure of entire pack. In other words, the goal is to find the balancing current u_k^n for each cell $n = 1, \ldots, N$

and for each time step $k = 0, \dots, K$ so that the cell voltage always satisfies $\underline{y} \le y_k^n$.

Note that u_k for all time steps k cannot be determined at the same time, since in practice we do not have the entire driving cycle i_k. Furthermore, solving u_k for all k together requires significant computational power that makes it impractical for real-time application. Therefore, we formulate the problem as a model predictive control (MPC) problem, which uses a relatively short horizon (compared to the full driving cycle) to predict the future evolution and optimizes the objective function over this prediction horizon. At each time step, a control sequence over the entire prediction horizon is obtained, but only the first element is implemented. At next time step, the whole process repeats. Denoting $u_k = [u_k^1, u_k^2, \dots, u_k^N]^T$, the optimal control problem (OCP) for MPC to solve at time k is given by

$$
\begin{aligned}
\min_{u_k} \quad & J(u_k) && \text{(7a)}\\
\text{s.t.} \quad & \delta\zeta_{k+j+1}^n = \hat{f}_k^n + A_k \delta\zeta_{k+j}^n + B_k \delta u_k^n + B d_k \delta i_{k+j}, \\
& \qquad\qquad 0 \le j \le p-1, 1 \le n \le N && \text{(7b)}\\
& y_{k+j}^n = \hat{g}_k^n + C_k \delta\zeta_{k+j}^n + D_k \delta u_k^n + D d_k \delta i_{k+j}, \\
& \qquad\qquad 1 \le j \le p, 1 \le n \le N && \text{(7c)}\\
& u_k^n = \hat{u}_k^n + \delta u_k^n, \qquad 1 \le n \le N && \text{(7d)}\\
& u_{\text{min}} \le u_k^n \le u_{\text{max}}, \qquad 1 \le n \le N && \text{(7e)}\\
& \underline{y} \le y_{k+j}^n, \qquad 1 \le j \le p, 1 \le n \le N && \text{(7f)}\\
& 0 = \sum_{n=1}^N u_k^n, && \text{(7g)}
\end{aligned}
$$

where p is the prediction horizon and $J(u_k)$ is the cost function to be formally defined in Section III-B. Constraints (7b) and (7c) are system dynamics given by (6). The last constraint (7g) indicates that the balancing circuit is only responsible to transport charge from one cell to another, and cannot provide or consume any additional charge.

Remark 1: Note that in OCP (7), the MPC is to optimize one balancing current u_k^n for each cell, which is then kept unchanged over the entire prediction horizon. This strategy is adopted from [4] as the balancing currents are almost constant over the prediction horizon. Such an arrangement can significantly reduce the size of the OCP for real-time implementation.

Remark 2: Note also that the above OCP (7) requires a short-term prediction of the load profile $i_{k+j}, 1 \le j \le p$ over the prediction horizon. If such preview is not available, the value at time k can be used throughout the whole horizon, i.e., $i_{k+j} = i_k, 1 \le j \le p$.

Remark 3: Note that though we consider constant lower bound $\underline{y}$ in (7f) for our numerical study, the OCP (7) can be straightforwardly extended to include the case when the lowest SOC/voltage is changing dynamically. In other words, the value of $\underline{y}$ can be changed dynamically to accommodate transient condition. In this case, a new value for $\underline{y}$ needs to be computed and received by MPC for each sampling time.

B. Objective Functions

Though the general balancing problem we consider is similar to that of [4], the cost function we consider for the OCP (7) will be much different from [4], resulting in completely different control strategies. In particular, we propose three different objective functions for (7a).

In the first formulation, we use a nominal cell model to integrate over the prediction horizon using the requested total current i_k (or i_{k+j} if preview is available), and the resulting voltage/SOC trajectory is set as reference for all cells to follow. The dynamics of the nominal cell are the same as those of (1) but with nominal parameters, as shown in Fig. 3. Then we integrate the nominal cell model using the initial condition $\zeta_k^0 = \frac{1}{N}\sum_{n=1}^N \zeta_k^n$ to obtain the nominal sequences $\zeta_{k+1}^0, \zeta_{k+2}^0, \dots, \zeta_{k+p}^0$ and $y_{k+1}^0, y_{k+2}^0, \dots, y_{k+p}^0$, and the cost function (7a) is defined as

$$
J_{\text{t},\sigma}(u_k) = \sum_{j=1}^{p} \left(\sigma_{k+j} - \sigma_{k+1}^0\right)^T \left(\sigma_{k+j} - \sigma_{k+j}{}^0\right) + u_k^T R u_k, \tag{8}
$$

where σ_{k+j} is defined as $\sigma_{k+j} = [\sigma_{k+j}^1, \sigma_{k+j}^2, \dots, \sigma_{k+j}^N]^T$ and $\sigma \in \{s, y\}$ can be either cell's SOC or terminal voltage, and R is a positive semi-definitive weighting matrix.

Remark 4: Note that the first term of (8) is to penalize the deviation from nominal trajectory while the second term prevents large balancing current that may result in energy waste through resistant heating. The first term does not require a scaling matrix as in this work the output being tracked is scalar, i.e., either SOC or voltage. The weighting among these two terms can be achieved through the R matrix alone.

Remark 5: In this work, the initial condition for integrating the nominal cell is given by averaging all cells' state vectors. Another approach is to utilize an observer to estimate the state of the nominal cell based on measurement from battery pack. This remains as future work.

The cost function (8) is intuitive to understand. However, the tracking of all cells can sometimes be too aggressive considering that the primary goal of balancing control is to ensure the minimum cell SOC/voltage stay away from the lowest bound. Therefore, we propose the second formulation which, instead of tracking a nominal trajectory, directly maximizes the lowest cell SOC or voltage. In other words, the cost function (7a) is defined as

$$
J_{\text{m},\sigma}(u_k) = -\sum_{j=1}^{p} \min_n \sigma_{k+j}^n + u_k^T R u_k, \tag{9}
$$

where σ_{k+j} is defined above. Note that the cost function $J_{\text{m},\sigma}$ is to *maximize* the lowest cell SOC/voltage for each time step over the prediction horizon with minimal balancing current. In order to reformulate $J_{\text{m},\sigma}$ to be manageable for embedded environment, the trick introduced in [28] is adopted as follows. With addition of p slack variables, $\epsilon = [\epsilon_1, \epsilon_2, \dots, \epsilon_p]^T$, the objective function (9) can be rewritten as,

$$
J_{\text{m},\sigma}(u_k, \epsilon) = -\sum_{j=1}^{p} \epsilon_j + u_k^T R u_k, \tag{10}
$$

with additional constraint

$$\epsilon_j \leq \sigma^n_{k+j}, \qquad 1 \leq j \leq p, \quad 1 \leq n \leq N. \tag{11}$$

Please refer to [28] for more details.

Finally, in the third (and last) formulation, instead of maximizing the lowest cell SOC or voltage, we set up the MPC to minimize the difference between the highest and lowest cell SOC/voltage. The main goal of this approach is to encourage MPC to move charges from cell with highest SOC/voltage. More specifically, the cost function (7a) is defined as

$$\begin{aligned} J_{\Delta,\sigma}(u_k) &= \sum_{j=1}^{p} \left(\max_n \sigma^n_{k+j} - \min_n \sigma^n_{k+j} \right) + u_k^T R u_k \\ &= \sum_{j=1}^{p} \max_n \sigma^n_{k+j} - \sum_{n=1}^{p} \min_n \sigma^n_{k+j} + u_k^T R u_k. \end{aligned} \tag{12}$$

Similarly, with a slight abuse of notation, define $2p$ slack variables, $\epsilon = [\epsilon_1, \epsilon_2, \ldots, \epsilon_p, \epsilon_{p+1}, \ldots, \epsilon_{2p}]^T$, and the objective function (12) can be rewritten as,

$$J_{\Delta,\sigma}(u_k, \epsilon) = \sum_{j=1}^{p} \epsilon_{p+j} - \sum_{j=1}^{p} \epsilon_j + u_k^T R u_k, \tag{13}$$

with additional constraint

$$\begin{aligned} \epsilon_j &\leq \sigma^n_{k+j}, \qquad 1 \leq j \leq p, 1 \leq n \leq N & \text{(14a)} \\ \epsilon_{p+j} &\geq \sigma^n_{k+j}, \qquad 1 \leq j \leq p, 1 \leq n \leq N. & \text{(14b)} \end{aligned}$$

Remark 6: Please note that in the last two formulations, constraints (11) and (14) are only one sided, e.g., $\epsilon_j \leq \sigma^n_{k+j}$ instead of $\epsilon_j \leq \pm\sigma^n_{k+j}$. This is due to the fact that both s^n and y^n are positive by design and hence the cost functions $J_{\mathrm{m},\sigma}$ and $J_{\Delta,\sigma}$ are not based on conventional infinity norm.

C. Complete MPC Setups

Putting everything together, the first MPC setup, denoted as $J_{\mathrm{t},\sigma}$, solves the following OCP

$$\begin{aligned} \min_{u_k} \quad & (8) \\ \text{s.t.} \quad & (7b), (7c), (7e), (7f), (7g). \end{aligned}$$

The second MPC setup, denoted as $J_{\mathrm{m},\sigma}$, solves the following OCP

$$\begin{aligned} \min_{u_k, \epsilon} \quad & (10) \\ \text{s.t.} \quad & (7b), (7c), (7e), (7f), (7g), (11). \end{aligned}$$

The third MPC setup, denoted as $J_{\Delta,\sigma}$, solves the following OCP

$$\begin{aligned} \min_{u_k, \epsilon} \quad & (13) \\ \text{s.t.} \quad & (7b), (7c), (7e), (7f), (7g), (14). \end{aligned}$$

Remark 7: Please note that all three MPC formulations can be cast into quadratic programming (QP) problem, which can be, in general, solved in real time by embedded devices [29], if the problem size is manageable. It is also worth noted that, $J_{\mathrm{t},\sigma}$ has $(2p+1)N$ optimization variables, $J_{\mathrm{m},\sigma}$ has $(2p+1)N+p$ optimization variables with additional pN constraints, while $J_{\Delta,\sigma}$ has $(2p+1)N+2p$ optimization variables with additional $2pN$ constraints. As can be seen, $J_{\mathrm{m},\sigma}$ and $J_{\Delta,\sigma}$ have larger problem sizes and require higher throughput to solve, while at the same time, provide certain benefits in some conditions, as will be seen in the next section.

Remark 8: Note that the output constraint (7f) can be infeasible when the cell voltage is approaching its lower bound. When this happens, we introduce an additional slack variable ϵ_y, and add to each cost function an additional term $W\epsilon_y^2$ where $W \gg R$, and modify the constraint (7f) into

$$\underline{y} \leq y^n_{k+j} + \epsilon_y, \qquad 1 \leq j \leq p, 1 \leq n \leq N. \tag{18}$$

In other words, MPC will solve the OCP with original constraint (7f), and when the OCP is found to be infeasible, MPC will then modify the cost function and replace (7f) with (18) as discussed here. Note that this is called "soft constraint" in literature, and has been applied to avoid infeasible OCP [29].

IV. Numerical Results

In this section, the effectiveness of the proposed MPC formulations will be demonstrated through simulations. Specifically, the linearized parametric model (6) will be used by MPC to form the OCPs, and the original nonlinear model (1) with SOC-dependent parameters and additive process noise will be used as simulation plant. Furthermore, two scenarios are considered. In the first scenario, a constant requested current i_k is considered, which is selected so that the simulation can be conducted in a reasonable amount of time. In the second scenario, the vehicle follows a realistic driving cycle, i.e., FTP cycle, where the vehicle is controlled by an MPC speed tracking controller that requested a battery power P_k [30]. At each time k, P_k is then converted to requested current by solving the quadratic equation as documented in [31]. Note that in this case, the preview of i_k is assumed to be unavailable, i.e., $\delta i_{k+j} = 0$ for (7b) throughout the entire prediction horizon. Due to the recent advancement of connected and automated vehicle, the preview of P_k may be estimated with acceptable noise. However, such availability assumption can be too restrictive for the present study.

For each of these two scenarios, the three MPC formulations with $\sigma = y$ will be considered. In other words, the first MPC (denoted as J_{t}) tracks all cells' voltages. The second MPC (denoted as J_{m}) maximizes the lowest cell voltage. And the third MPC (denoted as J_Δ) minimizes the difference between the highest and lowest cell voltages. For all setups, $N = 5$ is used and all cells are initialized to be fully charged. The cell parameters V^n_{oc}, R^n_o, R^n_p, and C^n_p are randomly generated to be within 10% deviation from the nominal values. Sampling time T_s is chosen to be 1 second, and $u_{\min} = -2\mathrm{A}$, $u_{\max} = 2\mathrm{A}$.

A. Constant Discharge Current

In this scenario, constant commanded current i_k is used to represent the steady state operation. Without active cell balancing, the battery pack can last 1,527 seconds until the lowest cell voltage drops below $\underline{y}$. For MPCs with prediction horizon $p = 5$, J_{t} can extend the operation time to 1,599

TABLE I
SIMULATION RESULTS FOR CONSTANT DISCHARGE CURRENT.

Setup	Operation Time [s]	Extension
No balancing	1,527	-
J_{t}	1,599	4.72%
J_{m}	1,640	7.40%
$J_{\triangle}$	1,628	6.61%

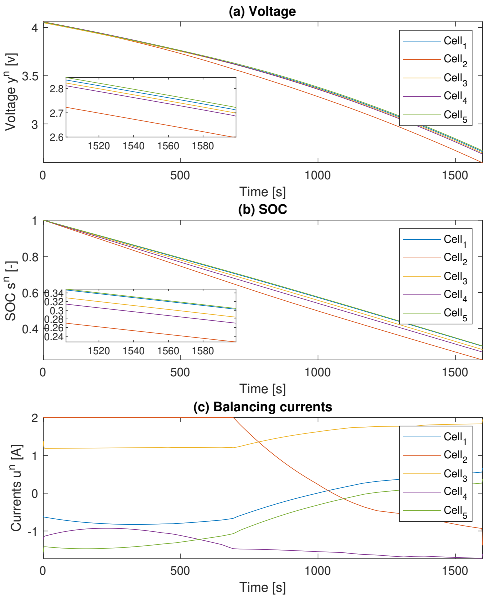

Fig. 4. Results for J_{t} with constant discharge current.

seconds (4.72% increase), J_{m} extends to 1,640 seconds (7.40% increase), while $J_{\triangle}$ extends to 1,682 seconds (6.61% increase). This is summarized in Table I and Figs. 4, 5 and 6, where each cell's voltage, SOC, and balancing current are plotted. It can be seen that the balancing currents for three MPCs possess a similar pattern, and are near constant or vary slowly for most of the time.

Furthermore, Fig. 7 compares the lowest cell voltage for different controllers, as well as the balancing effort, which represents an index for Ohmic heating loss due to balancing and is calculated as $e_k = u_k^T u_k$. It is clear from Fig. 7(b) that, J_{t} requires larger balancing efforts, especially when the SOC and voltage are still high. This is because J_{t} tracks all cell voltages to the nominal trajectory, and hence will try to balance even when all cell voltages are clear away from the lower bound $\underline{y}$. When the cell voltage gets closer to $\underline{y}$, all three MPCs utilize a similar amount of balancing efforts, while $J_{\triangle}$ is a little more aggressive.

To see the impact of prediction horizon, we set $p = 35$, reduce the current to a reasonable level, and at the same time scale the calibration R according to p to balance the two terms in the cost functions. For J_{t} formulation, without balancing, the battery failed at 5 hours, 16 minutes and 45 seconds, while with active cell balancing, it failed at 5 hours, 33 minutes and 52 seconds, providing a 5.13% range extension, which is a bit more than the 4.72% reported in Table I. Note that conducting a similar simulation for J_{m} and $J_{\triangle}$ is not possible due to the long simulation time (see Table III).

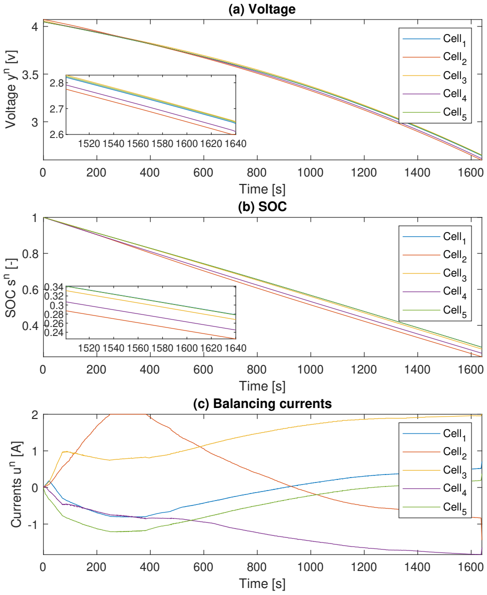

Fig. 5. Results for J_{m} with constant discharge current.

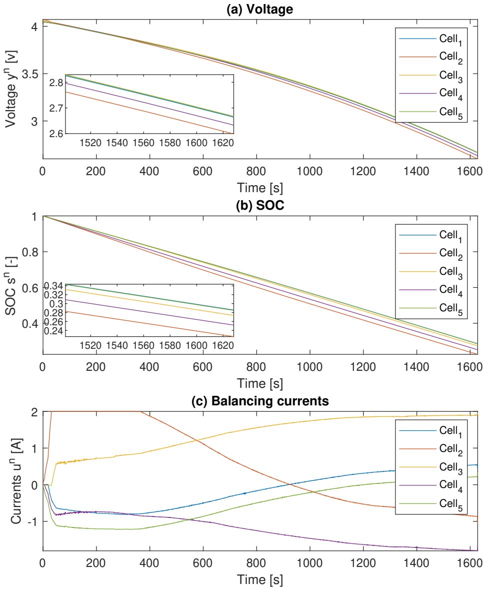

Fig. 6. Results for $J_{\triangle}$ with constant discharge current.

B. Realistic Driving Cycle

In this section, the vehicle follows a realistic driving cycle, i.e., FTP cycle, where the vehicle is controlled by an MPC speed tracking controller, as presented in [30]. The vehicle speed and power of FTP cycle is shown in Fig. 8, which is then concatenated and scaled up so to provide a realistic assessment of the range extension within a manageable amount of simulation time. In particular, over 3 hours and 20 minutes are simulated to mimic actual driving scenarios.

The range extensions for different controllers for $p = 5, 10, 15$, together with their balancing efforts defined as $e = \frac{1}{K}\sum_{k=1}^{K} e_k$, are presented in Tables II. With prediction horizon $p = 5$, all controllers can achieve 9.33% of range

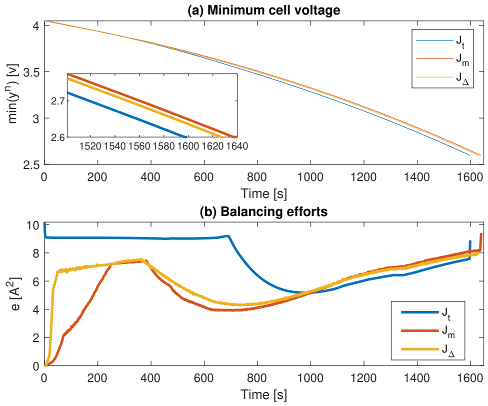

Fig. 7. Comparison of the lowest cell voltage and balancing efforts for different MPC formulations with constant discharge current.

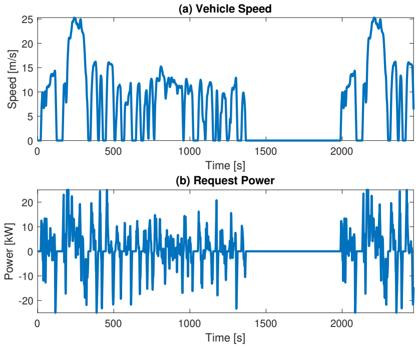

Fig. 8. Speed and power profiles of FTP.

extension, with very minimum balancing efforts. However, the range extensions slightly decrease with the increase of p. Such slight decrease may be due to the fact that we are not using preview on load profile in the present simulation, making longer prediction horizon less effective.

Finally, Fig. 9 plots the cell voltages for the three MPCs, where very similar behaviors are observed. Fig. 10 compares the minimum cell voltages and balancing efforts for a short period of time that is preceding to the pack failure. Though the minimum cell voltages for three MPCs are almost the same in Fig. 10(a), the balancing efforts are very much different. In particular, similar to the steady state scenario, J_t requires the maximum amount of balancing efforts.

TABLE II
SIMULATION RESULTS FOR FTP CYCLE.

Setup	p	Distance [m]	Extension	e [A^2]
No balancing	-	110949.0	-	-
J_t	5	121297.7	9.33%	3.8
J_m	5	121297.7	9.33%	0.4
$J_\triangle$	5	121297.7	9.33%	0.36
J_t	10	121297.7	9.33%	15.51
J_m	10	121297.7	9.33%	0.87
$J_\triangle$	10	120036.13	8.19%	2.7
J_t	15	120036.13	8.19%	15.71
J_m	15	119515.93	7.72%	8.61
$J_\triangle$	15	120036.13	8.19	11.68

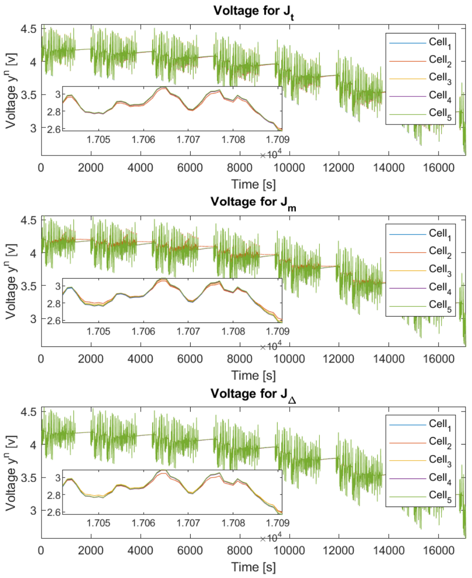

Fig. 9. Cell voltages comparison for different MPCs with FTP cycle.

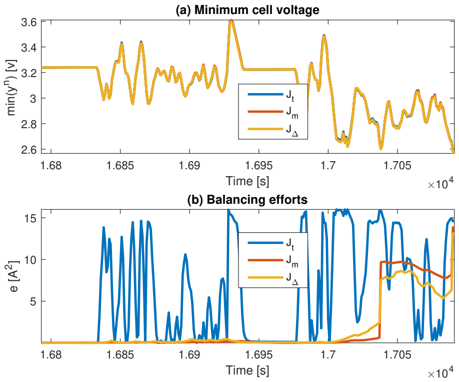

Fig. 10. Comparison of the lowest voltages and balancing efforts for different MPCs with FTP cycle.

C. Further Discussion

The throughputs required by each MPC are summarized in Table III, which is measured on a desktop computer with standard CPU using Matlab's standard matrix operations and quadprog as the QP solver. As can be seen, J_t is always manageable even for longer prediction horizon, while J_m and $J_\triangle$ are applicable for real time implementation only when p is smaller than 15.

From Table I, it can be seen that for steady state condition, J_m and $J_\triangle$ can achieve better range extensions with lower

TABLE III
THROUGHPUT (IN MILLISECOND) COMPARISON FOR DIFFERENT MPCS.

p	5	15	25	35
J_t	4.3	11.23	16.47	28.83
J_m	3.35	780.32	719.10	701.36
$J_\triangle$	8.34	2507.43	2128.6	5309.3

balancing efforts. However, they require a significant amount of throughput compared to J_t, according to Table III. In particular, the high throughput for $J_\triangle$ with long prediction horizon may prevent its real-time implementation in an embedded environment. Therefore, with shorter prediction horizon only, they seem to be better choices for steady state condition. On the other hand, according to Table II, for transient condition, J_t is much more robust against disturbance on future load profile, achieves better range extension with a slightly higher balancing efforts. Therefore, J_t seems to be a better choice for the transient condition.

V. Conclusions

In this paper, we studied the active cell balancing problem by using model predictive control (MPC) for electric vehicle range extension. Specifically, three MPC formulations were investigated. In the first formulation, a nominal cell was used to compute a short term reference trajectory and MPC was set to track all cell voltages to follow this reference trajectory. In the second and third formulations, MPC was set to maximize the lowest voltage cell and to minimize the difference between the highest and lowest cell voltage, respectively. In general, a 7% range extension can be achieved for steady state condition and 9% for trasident condition. It was also found that different driving scenarios may favor different MPC formulations, and a hybrid approach might be needed. Compared to the existing approaches in literature, our approach can achieve similar range extension without restrictively requiring the final battery state-of-charge to be known in advance. For future work, we would focus on (1) designing an observer to estimate the cells' voltage and SOC, (2) developing control algorithm to handle series-parallel connections, and (3) investigating the reason of difference performance in transient and steady state conditions.

References

[1] M. Kauer, S. Narayanaswamy, S. Steinhorst, M. Lukasiewycz, and S. Chakraborty, "Many-to-many active cell balancing strategy design," in *The 20th Asia and South Pacific Design Automation Conference*, Chiba, Japan, 2015, pp. 267–272.

[2] M. Dubarry, N. Vuillaume, and B. Y. Liaw, "Origins and accommodation of cell variations in li-ion battery pack modeling," *Int. J. of Energy Research*, vol. 34, no. 2, pp. 216–231, 2010.

[3] F. Hoekstra, H. J. Bergveld, and M. Donkers, "Range maximisation of electric vehicles through active cell balancing using reachability analysis," in *2019 American Control Conference (ACC)*, Philadelphia, PA, July 10-19, 2019, pp. 1567–1572.

[4] F. S. Hoekstra, L. W. Ribelles, H. J. Bergveld, and M. Donkers, "Real-time range maximisation of electric vehicles through active cell balancing using model-predictive control," in *2020 American Control Conference*, Denver, CO, July 1–3, 2020, pp. 2219–2224.

[5] K. Smith, E. Wood, S. Santhanagopalan, G. Kim, and A. Pesaran, "Advanced models and controls for prediction and extension of battery lifetime (presentation)," February 2014. [Online]. Available: https://www.osti.gov/biblio/1114881

[6] M. Einhorn, W. Roessler, and J. Fleig, "Improved performance of serially connected li-ion batteries with active cell balancing in electric vehicles," *IEEE Trans. Veh. Tech.*, vol. 60, no. 6, pp. 2448–2457, 2011.

[7] Y. Shang, B. Xia, C. Zhang, N. Cui, J. Yang, and C. C. Mi, "An automatic equalizer based on forward–flyback converter for series-connected battery strings," *IEEE Trans. Ind. Elect.*, vol. 64, no. 7, pp. 5380–5391, 2017.

[8] Y. Shang, N. Cui, and C. Zhang, "An optimized any-cell-to-any-cell equalizer based on coupled half-bridge converters for series-connected battery strings," *IEEE Transactions on Power Electronics*, vol. 34, no. 9, pp. 8831–8841, 2018.

[9] R. D. Anderson, R. Zane, G. Plett, D. Maksimovic, K. Smith, and M. S. Trimboli, "Life balancing–a better way to balance large batteries," SAE Technical Paper, Tech. Rep., 2017.

[10] M. Preindl, "A battery balancing auxiliary power module with predictive control for electrified transportation," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6552–6559, 2017.

[11] C. Wang, G. Yin, F. Lin, M. P. Polis, C. Zhang, J. Jiang *et al.*, "Balanced control strategies for interconnected heterogeneous battery systems," *IEEE Trans.n Sustainable Energy*, vol. 7, no. 1, pp. 189–199, 2015.

[12] Z. Gao, C. Chin, W. Toh, J. Chiew, and J. Jia, "State-of-charge estimation and active cell pack balancing design of lithium battery power system for smart electric vehicle," *Journal of Advanced Transportation*, 2017.

[13] S. Narayanaswamy, S. Park, S. Steinhorst, and S. Chakraborty, "Multi-pattern active cell balancing architecture and equalization strategy for battery packs," in *Proceedings of the International Symposium on Low Power Electronics and Design*, Seattle, WA, July 23–25, 2018, pp. 1–6.

[14] C. Pinto, J. V. Barreras, E. Schaltz, and R. E. Araujo, "Evaluation of advanced control for li-ion battery balancing systems using convex optimization," *IEEE Trans. Sustainable Energy*, vol. 7, no. 4, pp. 1703–1717, 2016.

[15] J. Liu, Y. Chen, and H. K. Fathy, "Nonlinear model-predictive optimal control of an active cell-to-cell lithium-ion battery pack balancing circuit," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 14 483–14 488, 2017.

[16] L. McCurlie, M. Preindl, and A. Emadi, "Fast model predictive control for redistributive lithium-ion battery balancing," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 1350–1357, 2016.

[17] Z. Gong, B. A. van de Ven, K. M. Gupta, C. da Silva, C. H. Amon, H. J. Bergveld, M. T. Donkers, and O. Trescases, "Distributed control of active cell balancing and low-voltage bus regulation in electric vehicles using hierarchical model-predictive control," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 12, pp. 10 464–10 473, 2019.

[18] J. Chen, A. Behal, and C. Li, "Active cell balancing by model predictive control for real time range extension," in *2021 IEEE Conference on Decision and Control*, Austin, TX, USA, December 13–15, 2021.

[19] Z. Pei, X. Zhao, H. Yuan, Z. Peng, and L. Wu, "An equivalent circuit model for lithium battery of electric vehicle considering self-healing characteristic," *Journal of Control Science and Engineering*, 2018.

[20] S. S. Madani, E. Schaltz, and S. Knudsen Kær, "An electrical equivalent circuit model of a lithium titanate oxide battery," *Batteries*, vol. 5, no. 1, p. 31, 2019.

[21] B. Y. Liaw, R. G. Jungst, A. Urbina, and T. L. Paez, "Modeling of battery life i. the equivalent circuit model (ECM) approach," Tech. Rep., 2003.

[22] H. He, R. Xiong, X. Zhang, F. Sun, and J. Fan, "State-of-charge estimation of the lithium-ion battery using an adaptive extended kalman filter based on an improved thevenin model," *IEEE Transactions on vehicular technology*, vol. 60, no. 4, pp. 1461–1469, 2011.

[23] J. Wehbe and N. Karami, "Battery equivalent circuits and brief summary of components value determination of lithium ion: A review," in *International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE)*. IEEE, 2015, pp. 45–49.

[24] M. Dubarry, V. Svoboda, R. Hwu, and B. Y. Liaw, "A roadmap to understand battery performance in electric and hybrid vehicle operation," *Journal of Power Sources*, vol. 174, no. 2, pp. 366–372, 2007.

[25] X. Lin, H. E. Perez, S. Mohan, J. B. Siegel, A. G. Stefanopoulou, Y. Ding, and M. P. Castanier, "A lumped-parameter electro-thermal model for cylindrical batteries," *Journal of Power Sources*, vol. 257, pp. 1–11, 2014.

[26] R. de Castro, C. Pinto, J. V. Barreras, R. E. Araújo, and D. A. Howey, "Smart and hybrid balancing system: Design, modeling, and experimental demonstration," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 11 449–11 461, 2019.

[27] T. Morstyn, M. Momayyezan, B. Hredzak, and V. G. Agelidis, "Distributed control for state-of-charge balancing between the modules of a reconfigurable battery energy storage system," *IEEE Transactions on Power Electronics*, vol. 31, no. 11, pp. 7986–7995, 2015.

[28] A. Alessio and A. Bemporad, "A survey on explicit model predictive control," in *Nonlinear model predictive control*. Springer, 2009, pp. 345–369.

[29] A. Bemporad, D. Bernardini, R. Long, and J. Verdejo, "Model predictive control of turbocharged gasoline engines for mass production," in *SAE Technical Paper*, 2018.

[30] J. Chen, M. Liang, and X. Ma, "Probabilistic analysis of electric vehicle energy consumption using mpc speed control and nonlinear battery model," in *2021 IEEE Green Tech. Conf.*, Denver, CO, April 7–9, 2021.

[31] Y. Kim, S. Mohan, J. B. Siegel, and A. G. Stefanopoulou, "Maximum power estimation of lithium-ion batteries accounting for thermal and electrical constraints," in *ASME Dynamic Systems and Control Conference*, 2013.