


Article

Comparison of Linear MPC and Explicit MPC for Battery Cell Balancing Control

Wanqun Yang and Jun Chen * 

Department of Electrical and Computer Engineering, Oakland University, Rochester, MI 48309, USA;
wanqunyang@oakland.edu

* Correspondence: junchen@oakland.edu

Abstract

This paper presents and compares two model predictive control (MPC) approaches for battery cell state-of-charge (SOC) balancing. In both approaches, a linearized discrete-time model that takes into account individual cell capacities is used. The first approach is a linear MPC controller that effectively regulates multiple cells to track a target SOC level while satisfying physical constraints. The second approach is based on explicit MPC implementation to reduce online computation while achieving a comparable performance. The simulation results suggest that explicit MPC can deliver the same balancing performance as linear MPC, while achieving faster online execution. Specifically, explicit MPC reduces the computation time by 37.3% in a five-cell battery example, with the cost of higher offline computation. However, simulation results also reveal a significant limitation for explicit MPC for battery systems with a larger number of cells. As the number of cells increases and/or the prediction horizon increases, the computational requirements grow exponentially, making its application to SOC balancing for large battery systems impractical. To the best of the authors' knowledge, this is the first study that compares MPC and explicit MPC algorithms in the context of battery cell balancing.

Keywords: active battery cell balancing; model predictive control; explicit model predictive control



Academic Editors: Van-Hai Bui,
Xuan Zhou, Wencong Su
and Akhtar Hussain

Received: 4 August 2025

Revised: 25 August 2025

Accepted: 25 August 2025

Published: 1 September 2025

Citation: Yang, W.; Chen, J.
Comparison of Linear MPC and
Explicit MPC for Battery Cell
Balancing Control. *Algorithms* **2025**,
18, 548. <https://doi.org/10.3390/a18090548>

Copyright: © 2025 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the Creative Commons
Attribution (CC BY) license
(<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The electric vehicle industry is regarded as one of the key measures to combat global warming [1]. However, the widespread adoption of lithium-Ion batteries, which is one of the main power sources for EVs, is still held back by issues such as state-of-charge (SOC) and voltage imbalance that can compromise driving range and slow down further development of the industry [2–4]. One of the primary causes of these imbalances lies in the cell-level characteristics within battery packs. Packs typically consist of multiple cells connected in series, but due to differences in manufacturing, temperature, internal resistance, and self-discharge rates, imbalance between cells gradually occurs [5,6]. This imbalance reduces the capacity of the entire pack, as the weakest cell limits the overall charge and discharge range. It also decreases efficiency, shortens battery lifespan, and may cause safety risks such as overvoltage or undervoltage in individual cells [7,8]. Therefore, to improve EV driving range, battery cell balancing control has been proposed to reduce differences among cells [7–10].

To overcome cell imbalance, two main balancing methods are commonly used: dissipative and redistributive. Dissipative balancing is simple but inefficient—it equalizes

cell voltages by discharging excess energy from higher-charged cells as heat through resistors, leading to energy loss [11]. In contrast, redistributive (or active) balancing uses power electronic circuits to transfer energy directly from cells with a higher SOC to those with a lower SOC, improving the energy efficiency and preserving the overall pack performance [12–14], using balancing circuits [15] such as flyback DC/DC converters [12] or half-bridge converters [16]. However, the redistributive balancing method requires advanced battery management systems to track and regulate cell SOC and voltage. In this paper, the redistributive method is primarily adopted.

The aim of active balancing is to prevent cell undervoltage by maintaining safe voltage levels, with many control methods explored in the literature. For example, ref. [17] discussed simple feedback control and refs. [18,19] implemented rule-based control to monitor and track SOC. Although these studies have shown encouraging results, they usually adopt simple control strategies that fall short in maximizing the utilization of energy stored in battery cells. The authors in [20,21] used heuristic control and the results appear to have been effective. In this paper, model predictive control (MPC), a real-time receding horizon control approach, is investigated as a more effective alternative. Specifically, we apply both linear MPC and explicit MPC for battery cell SOC balancing. The linear MPC relies on online computation to find the optimal balancing currents and requires only minimal memory, whereas the explicit MPC performs offline computation at the expense of significantly higher memory usage. Compared to existing MPC-based balancing [7,8,10,12], we focus on linear MPC to simplify the required onboard computation. Moreover, we provide a numerical analysis on the real-time implementability and scalability for large-scale battery packs. To the best of the authors' knowledge, this work is the first to focus on a computational comparison between MPC and explicit MPC in the context of battery SOC balancing. We firstly confirm that both linear MPC and explicit MPC can generate the same control inputs for battery balancing. Based on this, we investigate three factors that may affect the number of regions in explicit MPC: cell number, prediction horizon, and control horizon. The results provide numerical insights for future researchers on how to select appropriate cell numbers, prediction horizons, and control horizons when deploying explicit MPC for battery balancing. The results also offer practical insights into the selection of linear versus explicit MPC depending on whether the battery management system (BMS) hardware has more memory or a higher CPU performance.

Our contributions are summarized below.

1. A linear MPC controller is designed to regulate the SOC of battery cells to track a target SOC level.
2. An explicit MPC controller is also designed to perform the same balancing task, whose performance and computation requirements are compared with the linear MPC.
3. To provide insights on the scalability and real-time implementability of explicit MPC, an analysis is conducted on how the offline computation effort for the explicit MPC approach changes as the prediction horizon increases.

The remainder of this paper is organized as follows. Section 2 introduces system model and two MPC formulations, while the system settings and simulation results are presented in Section 3. Section 4 concludes the paper with future work directions.

2. Formulation

2.1. System Model

The serial-connected battery considered here is shown in Figure 1, where N cells are connected to provide current i to the load, e.g., an EV. We consider the conventional hierarchical topology, namely from cells to modules and from modules to pack, as the basis

for our simulations. We introduce the state-of-charge (SOC) variable $s_n(t) \in [0, 1]$ for the n -th cell. Its continuous-time dynamics under balancing current $i_n(t)$ (A) are

$$\dot{s}_n(t) = -\frac{1}{3600 C_n} i_n(t), \quad (1)$$

where C_n is the cell capacity in Ah; the factor 3600 converts the capacity from Amp-Hour to Amp-Second. Because manufacturing tolerances vary, each C_n may differ.

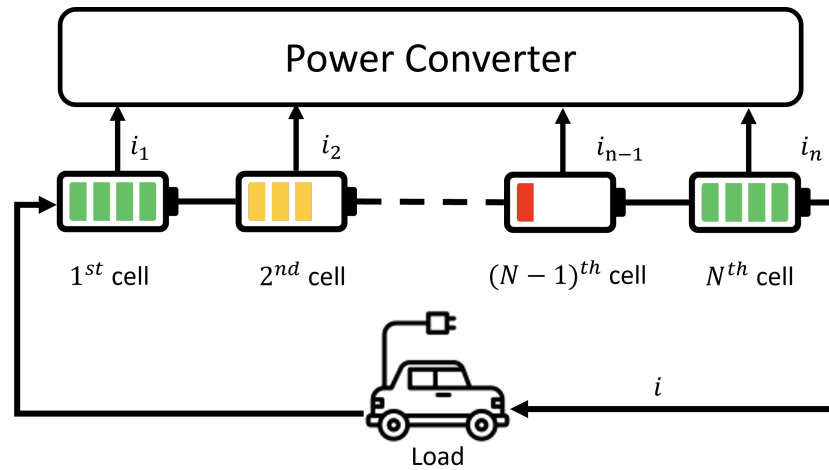


Figure 1. Structure of serial-connected battery cells.

Stacking all cell SOC and currents gives

$$x(s) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_N(t) \end{bmatrix}, \quad (2)$$

$$i(t) = \begin{bmatrix} i_1(t) \\ i_2(t) \\ \vdots \\ i_N(t) \end{bmatrix}. \quad (3)$$

With the diagonal matrix

$$M_c = -3600 \begin{bmatrix} C_1 & 0 & \cdots & 0 \\ 0 & C_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_N \end{bmatrix}, \quad (4)$$

the vector form of (1) is

$$\dot{x}(t) = M_c^{-1} i(t). \quad (5)$$

Using a sampling time T_s and forward Euler, we have

$$x_{k+1} = x_k + T_s M_c^{-1} i_k, \quad (6)$$

where i_k is the balancing current vector at time step k . It should be noted that this is different from the main discharge current i shown in Figure 1. Define

$$A = I, \quad (7a)$$

$$B = T_s M_c^{-1}, \quad (7b)$$

where I is the identity matrix, which we define as A to enhance readability and generalization of the model. So that

$$x_{k+1} = Ax_k + Bi_k. \quad (8)$$

For convenience in controller design, we shift SOC by \bar{s} , and with a slight abuse of notation:

$$x_k = \begin{bmatrix} s_1(k) - \bar{s} \\ s_2(k) - \bar{s} \\ \vdots \\ s_N(k) - \bar{s} \end{bmatrix}, \quad (9)$$

where \bar{s} denotes the target SOC level for all cells to track. Note that by using $s_n(k) - \bar{s}$ instead of $s_n(k)$ as state variables, we essentially reformulate the tracking control problem into a regulation problem, which can be easier for implementation.

With sampling time T_s and number of cells N , the matrices in (7) reduce to the following explicit forms

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}, \quad B = -T_s \cdot \begin{bmatrix} \frac{1}{3600C_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{3600C_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{3600C_N} \end{bmatrix} \quad (10)$$

Balancing currents are subject to magnitude limits and charge-conservation:

$$i_{\min} \leq i_n(k) \leq i_{\max}, \quad n = 1, \dots, N, \quad (11)$$

$$\sum_{n=1}^N i_n(k) = 0. \quad (12)$$

Constraint Equation (12) enforces overall charge conservation within the pack, meaning the sum of the balancing currents is zero. Note that this is simplified from the power conservation constraint and has been widely used in battery cell balancing literature [8].

Note that [22] implemented multiple estimation methods for lithium-ion batteries, and their results show that, after noise removal, the SOC dynamics are highly linear. Furthermore, since the battery balancing has been studied in literature [13,23–25] solely on simulation, our model is sufficient to meet the research objectives. Therefore, to better highlight the computational comparison between linear MPC and explicit MPC, we adopt a simplified battery model in this work.

2.2. Linear MPC

The linear MPC controller aims to regulate the SOC of all cells to a target value \bar{s} . At each time step, the controller solves a finite-horizon optimization problem to determine the optimal balancing current sequence. The optimization problem can be formulated as follows.

$$\min_u \quad \frac{1}{2} \sum_{k=1}^p \|x_k\|_{Q_x}^2 + \frac{1}{2} \sum_{k=0}^{p-1} \|u_k\|_{Q_u}^2 \quad (13a)$$

$$\text{s.t.} \quad \text{system dynamics (8), } k = 0, 1, \dots, p-1 \quad (13b)$$

$$\text{input constraints (11) and (12), } k = 0, 1, \dots, p-1 \quad (13c)$$

$$x_0 = \text{initial state,} \quad (13d)$$

where p is the prediction horizon, Q_x and Q_u are weight matrices for the states and inputs, respectively. The prediction horizon p is how many future steps the controller considers when optimizing. Q_x and Q_u are symmetric weight matrices that penalize state deviations and control effort; they are chosen to be diagonal in our simulations. For more mathematical definitions and derivations of MPC, readers are referred to [26]. u_k is the balancing current and i_k introduced in Section 2.1. Based on references [26–28] in the MPC field, we present the following derivation: The optimization problem (13) can be formulated as a quadratic programming (QP) problem,

$$\begin{aligned} \min_u \quad & \frac{1}{2} u^\top (H_x + M_u) u + f_x^\top u \\ \text{s.t.} \quad & \begin{cases} M_{AB} u \leq x_{\max} - M_{Ak} x_0 \\ -M_{AB} u \leq -x_{\min} + M_{Ak} x_0 \\ u \leq u_{\max} \\ -u \leq -u_{\min} \\ Du \leq \Delta_{\max} - r_u \\ -Du \leq -\Delta_{\min} + r_u \end{cases}, \end{aligned} \quad (14)$$

where

$$M_{AB} = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{p-1}B & A^{p-2}B & \cdots & B \end{bmatrix}, \quad (15)$$

$$M_{Ak} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^p \end{bmatrix}. \quad (16)$$

In the QP problem (13), x_{\max} and x_{\min} are the upper and lower bounds of the system states, while u_{\max} and u_{\min} are the upper and lower bounds of the control inputs, respectively. Δ_{\max} and Δ_{\min} represent the maximum and minimum changes in control inputs between two time steps. This setting is introduced to ensure more stable control inputs for the system. We denote

$$M_x = \begin{bmatrix} Q_x & 0 & 0 & \cdots & 0 \\ 0 & Q_x & 0 & \cdots & 0 \\ 0 & 0 & Q_x & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & Q_x \end{bmatrix}. \quad (17)$$

Moreover,

$$f_x = x_0^\top M_{Ak}^\top M_x M_{AB}, \quad (18)$$

$$M_u = \begin{bmatrix} Q_u & 0 & 0 & \cdots & 0 \\ 0 & Q_u & 0 & \cdots & 0 \\ 0 & 0 & Q_u & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & Q_u \end{bmatrix}, \quad (19)$$

$$r_u = \begin{bmatrix} -u_{-1} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (20)$$

2.3. Explicit MPC

For embedded applications with limited computational resources, explicit MPC provides an alternative approach by pre-computing the control law offline [29]. The explicit MPC approach for the same optimization problem transforms the online optimization into a function evaluation problem. The following mathematical derivation is also mainly inspired by [29]. We have

$$f_x = x_0^\top M_{Ak}^\top M_x M_{AB} = x_0^\top F, \quad (21)$$

It is worth noting that f_x and F are terms that are independent of u . They are intermediate variables introduced to simplify the mathematical derivation and improve readability. The above equation can be rewritten as a finite-horizon problem:

$$\begin{aligned} \min_u \quad & \frac{1}{2} u^\top (H_x + M_u) u + x_0^\top F u, \\ \text{subject to} \quad & \begin{cases} M_{AB} u \leq x_{\max} - M_{Ak} x_0, \\ -M_{AB} u \leq -x_{\min} + M_{Ak} x_0 \\ u \leq u_{\max} \\ -u \leq -u_{\min} \end{cases} \end{aligned} \quad (22)$$

Furthermore, denote

$$H = H_x + M_u \quad (23a)$$

$$G = \begin{bmatrix} M_{AB} \\ -M_{AB} \\ I \\ -I \end{bmatrix} \quad (23b)$$

$$W = \begin{bmatrix} x_{\max} \\ -x_{\min} \\ u_{\max} \\ -u_{\min} \end{bmatrix} \quad (23c)$$

$$S = \begin{bmatrix} -M_{Ak} \\ M_{Ak} \\ 0 \\ 0 \end{bmatrix}. \quad (23d)$$

where

$$H_x = M_{AB}^T M_x M_{AB}. \quad (24)$$

Equation (22) can be written as

$$\begin{aligned} \min_u \quad & \frac{1}{2} u^\top H u + x_0^\top F u \\ \text{subject to} \quad & G u \leq W + S x_0 \end{aligned} \quad (25)$$

the solution of this optimize problem is

$$H u^* + F^\top x_0 + G^\top \lambda^* = 0 \quad (26a)$$

$$\lambda_i^* (G^i u - W^i - S^i x_0) = 0, \quad i \in \mathcal{A} \quad (26b)$$

$$\lambda^* \geq 0 \quad (26c)$$

$$G u \leq W + S x_0 \quad (26d)$$

where \mathcal{A} is the set of active constraints, i.e., constraints that the equality holds, corresponding to the optimal solution \mathbf{u}^* . Then, the KKT condition above gives rise to the following linear system:

$$\begin{bmatrix} H & (G^{\mathcal{A}})^\top \\ G^{\mathcal{A}} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^* \\ \lambda^* \end{bmatrix} = \begin{bmatrix} -F^\top x_0 \\ W^{\mathcal{A}} + S^{\mathcal{A}} x_0 \end{bmatrix} \quad (27)$$

Since H is positive definite, we have

$$\mathbf{u}^* = -H^{-1} \left((G^{\mathcal{A}})^\top \lambda^* + F^\top x_0 \right). \quad (28)$$

Therefore,

$$\mathbf{u}^* = K(\mathcal{A}) x_0 + g(\mathcal{A}), \quad (29)$$

where

$$\begin{aligned} K(\mathcal{A}) = H^{-1} (G^{\mathcal{A}})^\top & \left(G^{\mathcal{A}} H^{-1} (G^{\mathcal{A}})^\top \right)^{-1} \\ & \cdot \left(S^{\mathcal{A}} + G^{\mathcal{A}} H^{-1} F^\top \right) - F^\top \end{aligned} \quad (30)$$

$$g(\mathcal{A}) = H^{-1} (G^{\mathcal{A}})^\top \left(G^{\mathcal{A}} H^{-1} (G^{\mathcal{A}})^\top \right)^{-1} W^{\mathcal{A}} \quad (31)$$

The optimal control to be implemented is given by

$$\begin{aligned} u^* &= [I \quad 0 \quad \cdots \quad 0] \mathbf{u}^* \\ &= [I \quad 0 \quad \cdots \quad 0] K(\mathcal{A}) x_0 + [I \quad 0 \quad \cdots \quad 0] g(\mathcal{A}) \end{aligned} \quad (32)$$

where $K(\mathcal{A})$ is a matrix whose first n_u rows define the gain K_A , and $g(\mathcal{A})$ is a vector whose first n_u entries define g_A . Then, we have the following affine feedback control law:

$$u^* = K_A x_0 + g_A. \quad (33)$$

The main advantage of explicit MPC is the reduction in online computational requirements, making it suitable for real-time implementation on embedded systems. However, as will be seen shortly, the offline computation increases exponentially with the state dimension and prediction horizon length, which is a significant limitation for high-dimensional systems. In terms of memory requirements, linear MPC computes optimal control online, so the information it stores grows polynomially with respect to the prediction horizon, whereas explicit MPC must store the control law and inequalities for many regions. As the

number of regions grows exponentially with respect to the prediction horizon, explicit MPC often needs much more memory.

3. Simulation Setup and Results

In this section, the numerical settings of the system along with the simulation results are presented. Both MPC controllers are applied to the same battery system.

3.1. System Settings

Both the linear and explicit MPC controllers are configured using the same system setup to ensure consistency and simplify comparison. The battery system consists of five cells, each with a capacity of 4.1 Ah. In many battery balancing studies, the initial imbalance deviations are larger. For example, ref. [30] performs balancing during charging, where the initial SOC difference between the maximum and minimum can reach 0.3. References [23,31] conduct balancing during discharging, with selected initial SOC deviations of about 0.1. Authors in [32] also provide an example under worst case conditions, where the initial SOC deviation reaches 0.45. In our simulation, initial SOC deviation is set to 0.2. The initial state x_0 is given by:

$$x_0 = \begin{bmatrix} -0.1 & -0.05 & 0.1 & 0.05 & 0 \end{bmatrix}^T, \quad (34)$$

with the target SOC level given by $\bar{s} = 0.5$. In other words, tracking x to 0 actually corresponds to tracking the SOC to 0.5, and in this case, the range of x is between -0.5 and 0.5 . The sampling time T_s is set to 20 s, and the total simulation duration is 500 min. For the weights, we test several different values in the linear MPC and select the set that yields a relatively smooth balancing current. The MPC weights are chosen as 10 for output tracking, 0.1 for input magnitude, and 0 for input rate, emphasizing accurate SOC regulation while minimizing control efforts. The values of i_{max} and i_{min} are 0.3 and -0.3 , respectively.

3.2. Simulation Results

Figures 2 and 3 present cell SOC deviations and balancing currents for the linear MPC controller. Specifically, Figure 2 presents results with a prediction horizon $P = 10$ and control horizon $m = 10$, while Figure 3 has a prediction horizon $P = 4$ and control horizon $m = 3$. As can be seen, it takes approximately 200 min to complete the battery SOC balancing. We can also observe that the system reaches balancing faster when the prediction horizon is longer. It should be noted that once the system finishes balancing, the power converter stops operating. This can also be observed from the balancing current trajectory in Figures 2–4; after the cell SOC returns to 0.5, the balancing current drops to 0.

Figure 4 presents the cell SOC deviations and balancing current for an explicit controller with a prediction horizon $P = 4$ and control horizon $m = 3$. Note that the weights for the cost function is same as for the linear MPC discussed above. It can be observed that the trend and converge time are the same as Figure 3 for the linear MPC, demonstrating that the explicit MPC can deliver the same performance, even though a large amount of computation is moved to offline.

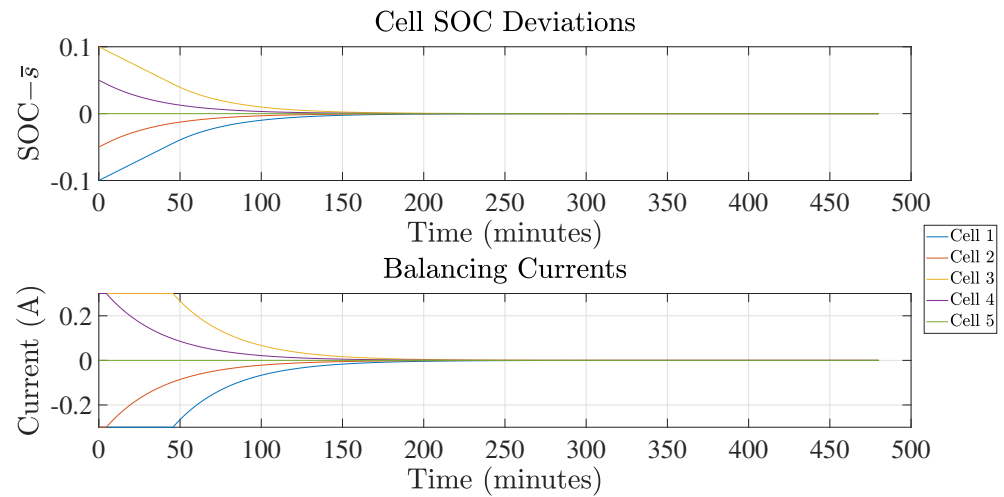


Figure 2. Simulation results for linear MPC on a battery pack with 5 serial-connected cells, with $p = m = 10$.

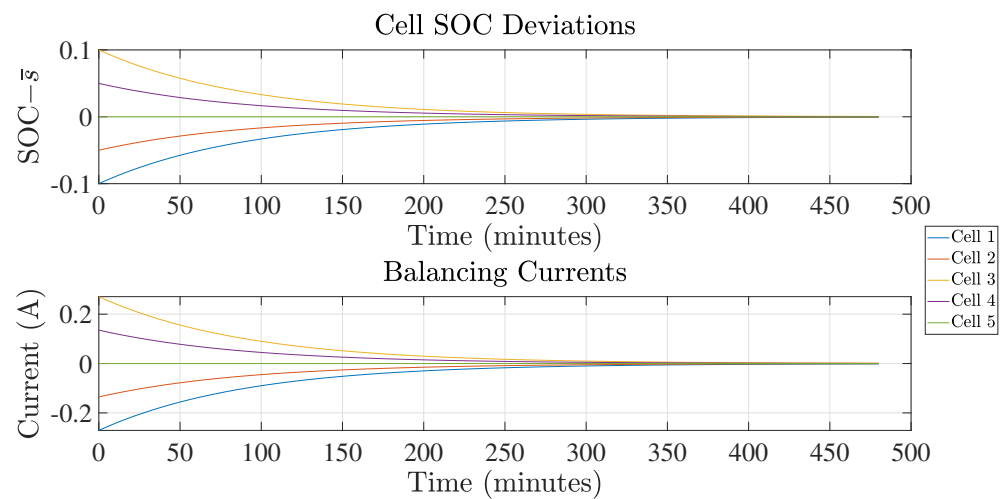


Figure 3. Simulation results for linear MPC on a battery pack with 5 serial-connected cells, with $p = 4$ and $m = 3$.

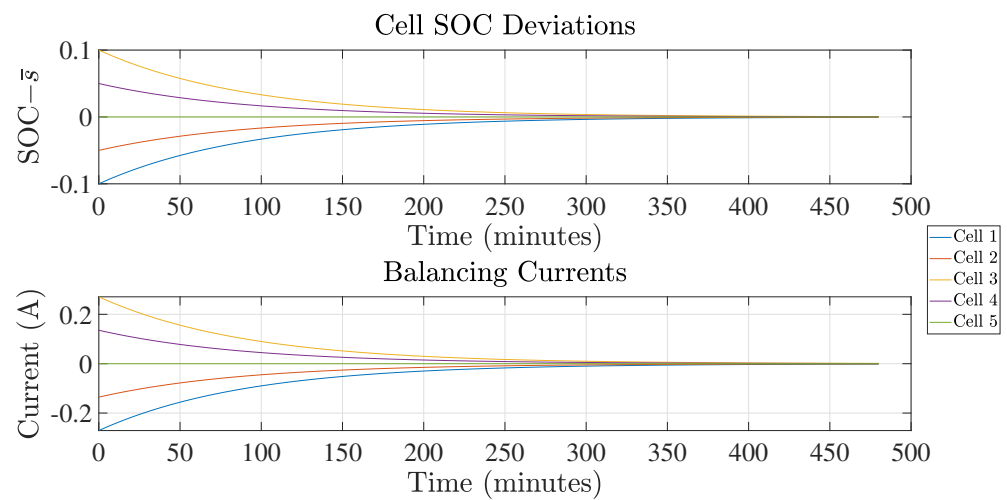


Figure 4. Simulation results for explicit MPC on a battery pack with 5 serial-connected cells, with $p = 4$ and $m = 3$.

3.3. Computation Analysis

The offline computation for explicit MPC increases exponentially with the prediction horizon due to the rapidly growing number of regions in the state-space partition. The processor of our computation platform is 13th Gen Intel(R) Core(TM) i9-13900F and for our 5-cell battery balancing problem with prediction horizon $p = 4$ and control horizon $m = 3$, the number of regions is 16,807 and the offline computation time is 502 s. For online computation, the explicit MPC requires 0.1637 s, while linear MPC (with the same p and m) requires 0.2610 s. It can then be observed that the online computation time decreases 37.3%.

When $p = 5$ and $m = 4$, the number of regions is 59,049, which is a huge computation burden. In fact, when $m = p$, the complexity follows approximately $O(2^{2(n_x+n_u)p})$, where n_x is the state dimension, n_u is the control input dimension, and p is the prediction horizon. In the cell balancing problem studied in this paper, $n_x = n_u = N$, where N is the number of cells in a battery pack. Then, we have that the complexity follows approximately $O(2^{4Np})$.

Table 1 shows the computation performance when $p = m = 3$ with the number of battery cells N ranging from 2 to 5, Figure 5 plots the number of regions and offline computation time with respect to N . In a lower dimension system where N equals to 2, 3, or 4, the offline computation time is negligible. When the N is equal to 5, the offline computation time rises to 8 min. When the N increases to 6, the offline computation time rises abruptly to about 115 min, which is 14.3 times larger than the case when $N = 5$. Table 2 and Figure 6 present the computation performance results when the system dimension remains unchanged and the prediction and control horizon increase gradually, which also confirm the exponential increase, but is slightly manageable compared with the number of battery cells. Since all systems use $N = 5$, each case is the same MPC problem with different prediction and control horizons. When $p = m = 2$ or 3, the online computation times are almost the same. As n and m increase, the online time also increases to some extent.

Table 1. Performance metrics for different number of cells ($p = m = 3$).

N	2	3	4	5	6
Regions	49	343	2401	16,807	117,649
Offline Time	0.079 s	1.747 s	30.927 s	479.361 s	6892.412 s
Online Time	0.109 s	0.114 s	0.115 s	0.116 s	0.637 s

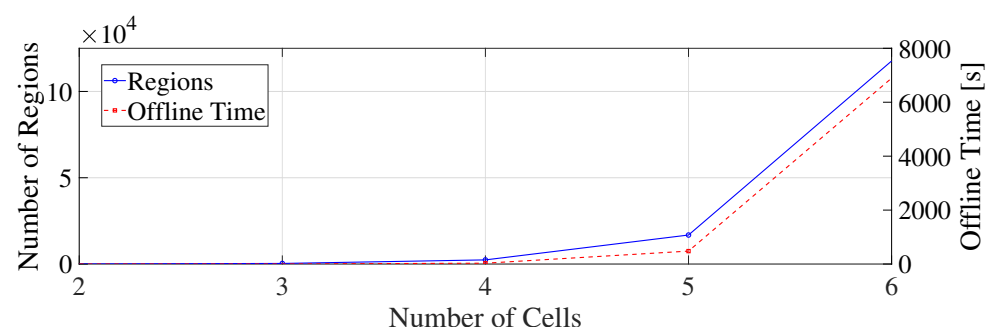


Figure 5. Regions and offline time vs. number of cells, with $p = m = 3$.

Table 2. Performance metrics for varying prediction horizons ($p = m$) with $N = 5$.

$p = m$	2	3	4	5	6
Region	3125	16,807	59,049	161,051	371,293
Offline Time	47.036 s	479.361 s	3122.997 s	15,869.483 s	60,357.946 s
Online Time	0.118 s	0.116 s	0.140 s	0.147 s	0.162 s

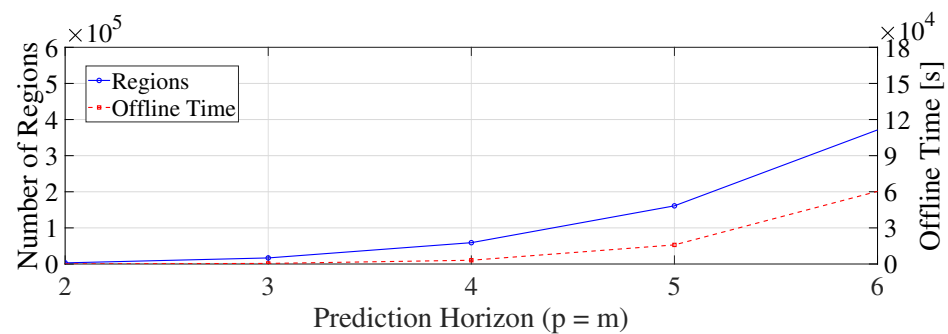


Figure 6. Regions and offline time vs. prediction and control horizon, with $N = 5$.

In Table 1, the memory usage of the five battery packs with different dimensions N is 9 KB, 37 KB, 287 KB, 2862 KB, and 26,528 KB. We can see that as the number of cells increases, the memory required for the explicit approach grows exponentially, indicating that deploying explicit MPC in high-dimension battery systems is not feasible. Finally, it is worth noting that the online computation time required by explicit MPC stays relatively stable when the system dimension changes. Therefore, if the offline computation is affordable with computing devices, such as high-performance computing clusters, then the real-time implementation should not be an issue for production-grade microcontrollers.

4. Conclusions

This paper explores two model predictive control (MPC) approaches, i.e., linear MPC and explicit MPC, for redistributive battery cell balancing in a serial-connected battery pack. A linearized discrete-time model is derived, taking into account individual cell capacities and enforcing charge conservation. Both controllers are implemented under identical system settings to ensure a fair comparison. Simulation results show that both MPC controllers successfully achieve SOC balancing. The linear MPC, although requiring online optimization at each time step, provides consistent performance with a moderate computational cost. In contrast, the explicit MPC achieves a faster online execution, reducing the onboard computation time by 37.3%, making it suitable for battery systems with a small number of cells. This set of experiments also highlights that, even though explicit MPC shifts most computations offline, the control performance remains comparable to that of linear MPC. However, explicit MPC incurs a substantial offline computational burden as the prediction horizon and/or number of cells increase. In fact, the exponential growth in offline efforts may limit the scalability of explicit MPC in battery systems with a large number of cells. The computation analysis notes that although both exhibit exponential growth, the increase in system dimension has a greater influence on the offline computation time compared to the increase in prediction and control horizons. Future work will scale up the simulation with more cells in the battery pack and explore methods to reduce the offline complexity of explicit MPCs, such as region merging or approximate controller synthesis. Moreover, we plan to improve the control strategy with further simulation validation and extend the model to a more realistic system by employing voltage-based SOC estimation. An experimental analysis will also be conducted to assess the performance of the current control design. Another direction is to develop a hybrid control framework that switches between explicit MPC and linear MPC to balance offline and online computation. In addition, we will investigate reduced-order MPC for this system and compare its performance against conventional explicit MPCs. Finally, building on these algorithm developments, we aim to deploy the controllers onto real hardware platforms for experimental validation.

Author Contributions: Conceptualization, J.C.; methodology, J.C.; software, W.Y.; validation, W.Y. and J.C.; formal analysis, W.Y. and J.C.; investigation, W.Y.; resources, J.C.; data curation, W.Y.; writing—original draft preparation, W.Y.; writing—review and editing, J.C.; visualization, W.Y.; supervision, J.C.; project administration, J.C.; funding acquisition, J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by the National Science Foundation through Award #2237317.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data are available upon request.

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

References

1. Lutsey, N. Global climate change mitigation potential from a transition to electric vehicles. *Int. Counc. Clean Transp.* **2015**, *2015*, 5.
2. Hu, X.; Yuan, H.; Zou, C.; Li, Z.; Zhang, L. Co-estimation of state of charge and state of health for lithium-ion batteries based on fractional-order calculus. *IEEE Trans. Veh. Technol.* **2018**, *67*, 10319–10329. [[CrossRef](#)]
3. Wang, Q.; Wang, Z.; Zhang, L.; Liu, P.; Zhang, Z. A novel consistency evaluation method for series-connected battery systems based on real-world operation data. *IEEE Trans. Transp. Electr.* **2020**, *7*, 437–451. [[CrossRef](#)]
4. Chen, J.; Zhou, Z.; Zhou, Z.; Wang, X.; Liaw, B. Impact of battery cell imbalance on electric vehicle range. *Green Energy Intell. Transp.* **2022**, *1*, 100025. [[CrossRef](#)]
5. Dubarry, M.; Vuillaume, N.; Liaw, B.Y. Origins and accommodation of cell variations in Li-ion battery pack modeling. *Int. J. Energy Res.* **2010**, *34*, 216–231. [[CrossRef](#)]
6. De Castro, R.; Pereira, H.; Araújo, R.E.; Barreras, J.V.; Pangborn, H.C. qTSL: A multilayer control framework for managing capacity, temperature, stress, and losses in hybrid balancing systems. *IEEE Trans. Control Syst. Technol.* **2021**, *30*, 1228–1243. [[CrossRef](#)]
7. Hoekstra, F.S.J.; Bergveld, H.J.; Donkers, M. Range maximisation of electric vehicles through active cell balancing using reachability analysis. In Proceedings of the 2019 American Control Conference (ACC), Philadelphia, PA, USA, 10–12 July 2019; pp. 1567–1572.
8. Hoekstra, F.S.J.; Ribelles, L.W.; Bergveld, H.J.; Donkers, M. Real-time range maximisation of electric vehicles through active cell balancing using model-predictive control. In Proceedings of the 2020 American Control Conference (ACC), Denver, CO, USA, 1–3 July 2020; pp. 2219–2224.
9. Chen, J.; Behal, A.; Li, C. Active Cell Balancing by Model Predictive Control for Real Time Range Extension. In Proceedings of the 2021 IEEE Conference on Decision and Control, Austin, TX, USA, 13–15 December 2021.
10. Chen, J.; Behal, A.; Li, Z.; Li, C. Active Battery Cell Balancing by Real Time Model Predictive Control for Extending Electric Vehicle Driving Range. *IEEE Trans. Auto. Sci. Eng.* **2024**, *21*, 4003–4015. [[CrossRef](#)]
11. Anderson, R.D.; Zane, R.; Plett, G.; Maksimovic, D.; Smith, K.; Trimboli, M.S. *Life Balancing—A Better Way to Balance Large Batteries*; Technical Report, SAE Technical Paper; SAE International: Warrendale, PA, USA, 2017.
12. Einhorn, M.; Roessler, W.; Fleig, J. Improved performance of serially connected Li-ion batteries with active cell balancing in electric vehicles. *IEEE Trans. Veh. Technol.* **2011**, *60*, 2448–2457. [[CrossRef](#)]
13. Daowd, M.; Omar, N.; Van Den Bossche, P.; Van Mierlo, J. Passive and active battery balancing comparison based on MATLAB simulation. In Proceedings of the 2011 IEEE Vehicle Power and Propulsion Conference, Chicago, IL, USA, 6–9 September 2011; pp. 1–7.
14. Mestrallet, F.; Kerachev, L.; Crebier, J.C.; Collet, A. Multiphase interleaved converter for lithium battery active balancing. *IEEE Trans. Power Electron.* **2013**, *29*, 2874–2881. [[CrossRef](#)]
15. Shang, Y.; Xia, B.; Zhang, C.; Cui, N.; Yang, J.; Mi, C.C. An automatic equalizer based on forward-flyback converter for series-connected battery strings. *IEEE Trans. Ind. Electron.* **2017**, *64*, 5380–5391. [[CrossRef](#)]
16. Nie, J.; Fu, R.; Cai, C.; Ma, J.; Shu, Z.; Ma, L. A high efficiency battery equalizing circuit based on half bridge topology with multiport transformer. *IEEE Trans. Ind. Electron.* **2023**, *71*, 2522–2532. [[CrossRef](#)]
17. Wang, C.; Yin, G.; Lin, F.; Polis, M.P.; Zhang, C.; Jiang, J. Balanced control strategies for interconnected heterogeneous battery systems. *IEEE Trans. Sustain. Energy* **2015**, *7*, 189–199. [[CrossRef](#)]
18. Xu, J.; Cao, B.; Li, S.; Wang, B.; Ning, B. A hybrid criterion based balancing strategy for battery energy storage systems. *Energy Procedia* **2016**, *103*, 225–230. [[CrossRef](#)]
19. Gao, Z.; Chin, C.; Toh, W.; Chiew, J.; Jia, J. State-of-charge estimation and active cell pack balancing design of lithium battery power system for smart electric vehicle. *J. Adv. Transp.* **2017**, *2017*, 6510747. [[CrossRef](#)]

20. Ouyang, Q.; Han, W.; Zou, C.; Xu, G.; Wang, Z. Cell balancing control for lithium-ion battery packs: A hierarchical optimal approach. *IEEE Trans. Ind. Inform.* **2019**, *16*, 5065–5075. [[CrossRef](#)]
21. Narayanaswamy, S.; Park, S.; Steinhorst, S.; Chakraborty, S. 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, USA, 23–25 July 2018; pp. 1–6.
22. Meng, J.; Ricco, M.; Luo, G.; Swierczynski, M.; Stroe, D.I.; Stroe, A.I.; Teodorescu, R. An overview and comparison of online implementable SOC estimation methods for lithium-ion battery. *IEEE Trans. Ind. Appl.* **2017**, *54*, 1583–1591. [[CrossRef](#)]
23. Ren, H.; Zhao, Y.; Chen, S.; Wang, T. Design and implementation of a battery management system with active charge balance based on the SOC and SOH online estimation. *Energy* **2019**, *166*, 908–917. [[CrossRef](#)]
24. Petri, A.M.; Petreus, D. Balancing and SOC Estimation in a Battery Management System for Electric Vehicle. In Proceedings of the 2021 44th International Spring Seminar on Electronics Technology (ISSE), Bautzen, Germany, 5–9 May 2021; pp. 1–6.
25. Yildirim, B.; Elgendy, M.; Smith, A.; Pickert, V. Evaluation and comparison of battery cell balancing methods. In Proceedings of the 2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), Bucharest, Romania, 29 September–2 October 2019; pp. 1–5.
26. Rawlings, J.B.; Mayne, D.Q.; Diehl, M.M. *Model Predictive Control: Theory, Computation, and Design*; Nob Hill Publishing: Madison, WI, USA, 2020; Volume 2.
27. Bishop, R.C.D.R.H. *Modern Control Systems*; Pearson Education, Inc.: London, UK, 2011.
28. Chen, C.T. *Linear System Theory and Design*; Saunders College Publishing: Philadelphia, PA, USA, 1984.
29. Bemporad, A.; Morari, M.; Dua, V.; Pistikopoulos, E.N. The explicit linear quadratic regulator for constrained systems. *Automatica* **2002**, *38*, 3–20. [[CrossRef](#)]
30. Zhang, Z.; Zhang, L.; Hu, L.; Huang, C. Active cell balancing of lithium-ion battery pack based on average state of charge. *Int. J. Energy Res.* **2020**, *44*, 2535–2548. [[CrossRef](#)]
31. Cao, Y.; Qahouq, J.A.A. Hierarchical SOC balancing controller for battery energy storage system. *IEEE Trans. Ind. Electron.* **2020**, *68*, 9386–9397. [[CrossRef](#)]
32. Chauhan, S.R.; Kumar, K.; Nadarajan, S.; Vaiyapuri, V.; Halick, M.; Sathik, M. Effect of Unbalanced Cells in Lithium-ion Battery Pack Performance and SOC Estimation. In Proceedings of the 2024 12th International Conference on Internet of Everything, Microwave, Embedded, Communication and Networks (IEMECON), Jaipur, India, 24–26 October 2024; pp. 1–6.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.