MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Distributed Optimal Power Management for Battery Energy Storage Systems: A Novel Accelerated Tracking ADMM Approach

Farakhor, Amir; Wang, Yebin; Wu, Di; Fang, Huazhen

TR2023-054 May 31, 2023

Abstract

Optimal power management (OPM) is critical for large-scale battery energy storage systems. Today's methods often require formidable computational effort due to the design based on centralized numerical optimization. Thus, this paper investigates computationally distributed OPM where the agents based on the cells communicate over a network to cooperatively solve the OPM problem. We propose an accelerated tracking alternating direction method of multipliers (ADMM) algorithm to solve the distributed OPM. The proposed algorithm em- beds dynamic average consensus and Nesterov's acceleration technique in the ADMM algorithm. Not only is the proposed algorithm fully distributed without a need for fusion or aggregating nodes, but it also accelerates the convergence. The paper formulates the OPM in a model predictive control framework where it seeks to regulate the charging/discharging power of each battery cell to minimize the total power losses and promote balanced use of the constituent cells while complying with the safety constraints. The paper provides ample simulation results to demonstrate the effectiveness and advantages of the proposed distributed OPM in terms of computation and convergence.

American Control Conference (ACC) 2023

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Distributed Optimal Power Management for Battery Energy Storage Systems: A Novel Accelerated Tracking ADMM Approach

Amir Farakhor[1], Yebin Wang[2], Di Wu[3] and Huazhen Fang[1]

***Abstract*— Optimal power management (OPM) is critical for large-scale battery energy storage systems. Today's methods often require formidable computational effort due to the design based on centralized numerical optimization. Thus, this paper investigates computationally distributed OPM where the agents based on the cells communicate over a network to cooperatively solve the OPM problem. We propose an accelerated tracking alternating direction method of multipliers (ADMM) algorithm to solve the distributed OPM. The proposed algorithm embeds dynamic average consensus and Nesterov's acceleration technique in the ADMM algorithm. Not only is the proposed algorithm fully distributed without a need for fusion or aggregating nodes, but it also accelerates the convergence. The paper formulates the OPM in a model predictive control framework where it seeks to regulate the charging/discharging power of each battery cell to minimize the total power losses and promote balanced use of the constituent cells while complying with the safety constraints. The paper provides ample simulation results to demonstrate the effectiveness and advantages of the proposed distributed OPM in terms of computation and convergence.**

I. INTRODUCTION

Large-scale battery energy storage systems (BESS) have found wide use in various sectors to enable applications such as electric vehicles, electric aircraft, and grid-scale energy storage. They comprise a large number of battery cells and a battery management system (BMS) to regulate the charging and discharging of the cells to guarantee their safe and reliable operation. Conventional BMS algorithms are often too simplistic to extract the full potential of BESS. Thus arises a pressing need to develop advanced BMS algorithms to achieve sophisticated functions. A provenly useful approach to this end is to enable independent power management at the level of individual cells within BESS. A few optimal power management (OPM) algorithms have emerged in this regard. However, they often require hefty computational efforts due to the use of numerical optimization techniques, making them hardly applicable especially for large-scale BESS. While the literature includes many OPM studies for different types of BESS, few of them have attempted to improve the computational efficiency of OPM.

The OPM problem for BESS encompasses a set of key issues, including but not limited to cell balancing, power loss minimization, and charging control. The crucial role that it plays in BESS operation has made it an appealing subject of study. Early studies usually focus on cell balancing based on optimization. The work in [1] employs linear programming to balance the state-of-charge of the cells of a pack with either minimum time or minimum energy dissipation. The notion is then extended in [2] to achieve two-level balancing, i.e., cell-to-cell balancing at intra-module level, and module-to-module balancing at inter-module level. The study in [3] identifies that a circuit designed for SoC balancing can potentially offer more functions, if using converters with bidirectional power flow control. This leads to the formulation of a multi-objective convex optimization problem in [3] to optimally balance the cells in terms of SoC, terminal voltage equalization, and temperature distribution. The method is further improved in [4] by introducing power loss minimization in the problem formulation. It further finds utility in enabling the OPM for a BESS in [5] based on a hybridization of cells, supercapacitors, and converters. Extending [5], the work in [6] presents a multi-layer model predictive control (MPC) framework to divide OPM tasks into two layers running at different time scales for the sake of less computation. Another important dimension of the OPM problem, BESS charging control also admits solutions from the perspective of optimization. Multi-objective optimization is exploited in [7] to enable user-defined charging under cell balancing and temperature constraints.

In hindsight, the existing OPM approaches harness the power of optimization to bring about a few valuable functions for BESS for better performance, safety, and longevity. However, numerical optimization at their basis usually requires many computational resources. The computational demand will reach a formidable level when a large-scale BESS imposes a large number of optimization variables. Even though the literature presents some hierarchical frameworks to alleviate the issue, there is still a research gap toward BESS OPM with high computational efficiency.

Distributed control has proven as a useful paradigm for control of large-scale systems. It distributes a control task and consequently, the computation, among the constituent units (often referred to as agents) of a system. This hence leads to high computational efficiency and scalability. The idea has found its way into BESS power control. For example, the studies in [8, 9] views the cells or modules constituting a BESS as independent agents, and then leverages the concept of distributed average consensus of networked multi-agent systems to design SoC balancing algorithms. Computationally fast as they are, these methods, however, lack optimality by design and thus are unable to optimize some important metrics, e.g., power losses, for BESS. Distributed

[1]A. Farakhor and H. Fang are with the Department of Mechanical Engineering, Univeristy of Kansas, Kansas, Lawrence, USA. a.farakhor@ku.edu, fang@ku.edu

[2]Y. Wang is with the Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. yebinwang@merl.com

[3]D. Wu is with the Pacific Northwest National Laboratory, Richland, WA, USA. di.wu@pnnl.gov

optimization holds a promise to overcome this limitation but has never been explored for the BESS OPM problem, even though it has been used for coordination of distributed energy resources (DERs), e.g., [10–12].

This paper proposes to harness the power of distributed optimization to achieve computationally viable OPM of large-scale BESS. We first formulate the OPM problem from the perspective of distributed optimization. In the formulation, the cells are treated as agents, and they perform individual computation to collectively minimize a global cost function under local constraints. Specifically, the global cost is set as the total power loss, and local constraints derive from the dynamics of the cells, SoC and temperature balancing requirements, and safety limits. We then propose an accelerated tracking alternating direction method of multipliers (ADMM) to solve the distributed OPM problem. The proposed accelerated ADMM algorithm is fully distributed and provides an improved convergence rate. We conduct simulations to show the effectiveness of the proposed algorithm. We also compare the obtained results with a previously proposed algorithm to demonstrate its advantages in computation and convergence.

II. OPTIMAL POWER MANAGEMENT OF BESS

This section describes the considered circuit structure of a large-scale BESS, its electro-thermal modeling, and the OPM formulation.

A. Circuit Structure of a Large-Scale BESS

Fig. 1 depicts the circuit structure of the considered large-scale BESS, which was first introduced by the authors in [13, 14]. The BESS comprises n cells, with each cell connected with a converter. We refer to such a cell-converter pair as a module. The modules are assumed to be connected in series here, even though the connection between them can be made reconfigurable as shown in [14]. The converters allow bidirectional power conversion to charge and discharge the cells. They hence can take the role of independently managing the power of the cells to provide the capability of cell-level control. The capability lays a basis for various advanced functions, including power loss minimization and cell balancing. In this paper, we leverage the structure to achieve OPM for the BESS and particularly, focus on enabling distributed optimization among the modules for high computational efficiency.

B. Electro-thermal Modeling

To begin with, we seek to characterize the electrical dynamics of each module of the BESS. We use the Rint model to represent a cell, which comprises a voltage source and an internal series resistor. The DC/DC converters are also modeled by an ideal DC transform in series connection with a resistor. We illustrate the module model in Fig. 1. Considering cell j, the governing equations of the Rint model are:

$$\dot{q}_j(t) = -\frac{1}{\bar{Q}_j} i_j(t), \tag{1a}$$

$$v_j(t) = u_j(q_j(t)) - R_j i_j(t), \tag{1b}$$

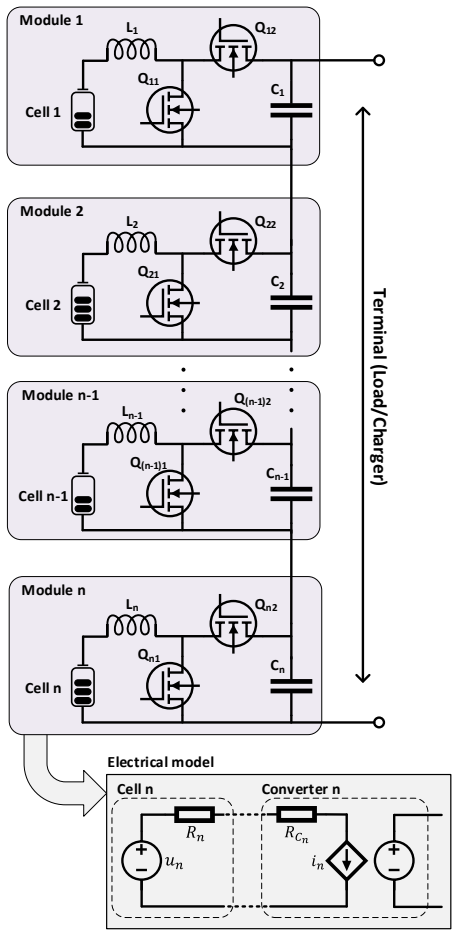


Fig. 1: The circuit structure of large-scale BESS.

where v_j, u_j, R_j, i_j, $\bar{Q}_j$, q_j are the terminal voltage, open-circuit voltage (OCV), internal resistance, current, capacity, and SoC, respectively. The output power of module j can be expressed as

$$P_j = u_j(q_j(t))i_j(t) - R_j i_j^2(t) - R_{C_j} i_j^2(t), \tag{2}$$

where $R_j i_j^2(t)$ and $R_{C_j} i_j^2(t)$ represent the internal power losses of cell j and converter j, respectively. The described electrical model is sufficiently expressive and computationally amenable.

Next, we can proceed to the thermal dynamics of the cells. We assume a lumped thermal capacitance for a cell, and the cell is subject to two effects on its temperature: heat generation caused by the power losses due to the cells' internal resistors, and heat dissipation to the environment due to convection. Considering cell j, the thermal dynamics is given by

$$C_{\text{th},j}\dot{T}_j(t) = R_j i_j^2(t) - (T_j(t) - T_{\text{env}})/R_{\text{conv}}, \tag{3}$$

where $C_{\text{th},j}$, T_j, T_{env}, and R_{conv} are the thermal capacitance, cell's temperature, environmental temperature, and convective thermal resistance, respectively. With the electro-thermal models for all the cells, we are ready to formulate the OPM problem for the BESS.

C. OPM Problem Formulation

We consider an OPM problem to minimize the power losses of the BESS while promoting the balanced use of the cells and complying with the safety constraints. In this regard, we consider predictive optimization over a receding horizon and define the cost function as

$$J(t) = \int_{t}^{t+H} \left(\sum_{j=1}^{n} (R_j + R_C) i_j^2(t) \right) dt, \tag{4}$$

where H is the horizon length. Further, the following constraints guarantee the safe operation of the BESS.

$$i_j^{\min} \leq i_{L_j} \leq i_j^{\max}, \tag{5a}$$
$$q_j^{\min} \leq q_j \leq q_j^{\max}, \tag{5b}$$

where $i_j^{\min/\max}$ and $q_j^{\min/\max}$ represent the lower/upper bounds for the cells' current and SoC. We also impose two balancing constraints in terms of SoC and temperature to encourage the balanced use of the cells.

$$|q_j(t) - q_{\text{avg}}(t)| \leq \Delta q, \tag{6}$$
$$|T_j(t) - T_{\text{avg}}(t)| \leq \Delta T, \tag{7}$$

where $q_{\text{avg}}(t)$ and $T_{\text{avg}}(t)$ are the average SoC and temperature of the BESS, and Δq and ΔT are the maximum tolerated deviation of the SoC and temperature from the respective average. To meet the output power demand, we enforce the following power conservation constraint:

$$\sum_{j=1}^{n} P_j = P_{\text{out}}, \tag{8}$$

where P_{out} is the output power demand. Putting together the cost function and the constraints, we express the OPM problem as follows:

$$\begin{aligned} \min_{i_j, j=1,\ldots,n} \quad & J(t), \\ \text{s.t.} \quad & \text{(1b)}, (3), (5a) - (8). \end{aligned} \tag{9}$$

Note that the optimization problem in (9) is non-convex due to the power conservation constraint (8). The solution is non-trivial and computationally expensive. We address this issue by a slight relaxation of the problem to make it convex, as suggested in [15].

D. Convex Formulation of OPM Problem

Here, we briefly discuss the convex formulation of the OPM problem (see [5] for more details). First, we assume a piecewise linear approximation for the SoC/OCV curve of cell j.

$$u_j(q_j(t)) = \alpha_j^i + \beta_j^i q_j(t), \tag{10}$$

where α_j^i and β_j^i define the y-intercept and slope of the ith line segment of cell j. The convexification proceeds with a change of variables. We define the cell's energy as follows:

$$E_j(t) = \frac{1}{2} C_j u_j^2(q_j(t)) - E_j^0, \tag{11}$$

where E_j^0 is the cell's initial energy, and $C_j = \bar{Q}_j / \beta_j$. Substituting (10) into (11) and using (1b), one can find that the cell's energy is governed by a linear dynamic equation:

$$\dot{E}_j(t) = -P_{b_j}, \tag{12}$$

where $P_{b_j}(t) = u_j(q_j(t)) i_j$ is the internal power of cell j. The power loss of module j is also expressed in terms of P_{b_j} as follows:

$$P_{l_j}(t) = \frac{(R_j + R_{C_j}) C_j P_{b_j}^2(t)}{2(E_j(t) + E_j^0)}. \tag{13}$$

Since our goal is to minimize the power losses, we can relax (13) for the sake of convexity as follows:

$$P_{l_j}(t) \geq \frac{(R_j + R_C) C_j P_{b_j}^2(t)}{2(E_j(t) + E_j^0)}. \tag{14}$$

Finally, we can rewrite the safety, balancing, and power conservation constraints (5)-(8) in terms of E_j, P_{b_j}, and P_{l_j}, and further apply discretization with a step size of Δt to obtain the following discrete-time convex OPM problem:

$$\min_{P_{b_j}, j=1,\ldots,n} \quad \sum_{j=1}^{n} \sum_{k=1}^{H} P_{l_j}^2[k] + \lambda^{(E)} {\xi_j^{(E)}}^2[k] + \lambda^{(T)} {\xi_j^{(T)}}^2[k],$$

Safety constraints:

$$\sqrt{\frac{2}{C_j}(E_j[k] + E_j^0)} i_{L_j}^{\min} \leq P_{b_j}[k] \leq \sqrt{\frac{2}{C_j}(E_j[k] + E_j^0)} i_{L_j}^{\max},$$
$$\frac{1}{2} C_j u_j^2(q_j^{\min}[k]) \leq E_j[k] + E_j^0 \leq \frac{1}{2} C_j u_j^2(q_j^{\max}[k]),$$

Balancing constraints:

$$\left| \frac{2}{C_j} E_j[k] - \frac{1}{n} \sum_{l=1}^{n} \frac{2}{C_l} E_l[k] \right| \leq \Delta E_j + \xi_j^{(E)}[k],$$
$$|T_j[k] - T_{\text{avg}}[k]| \leq \Delta T + \xi_j^{(T)}[k],$$

Power loss constraint:

$$P_{l_j}[k] \geq \frac{(R_j + R_C) C_j P_{b_j}^2[k]}{2(E_j[k] + E_j^0)},$$

Energy dynamics:

$$E_j[k+1] - E_j[k] = -P_{b_j}[k] \Delta t,$$

Thermal dynamics:

$$T_j[k+1] = T_j[k] + \frac{\Delta t}{C_{\text{th},j}} \Big[P_{l_j}[k] - (T_j[k] - T_{\text{env}}) / R_{\text{conv}} \Big],$$

Power supply-demand balance:

$$\sum_{j=1}^{n} P_{b_j}[k] - P_{l_j}[k] = P_{\text{out}}[k]. \tag{15}$$

In (15), we have also relaxed the balancing constraints by introducing the slack variables $\xi^{(E)}$ and $\xi^{(T)}$. This helps prevent the optimization problem from becoming infeasible when the balancing constraints cannot be satisfied. The penalty weights $\lambda^{(E)}$ and $\lambda^{(T)}$ associated with the slack variables in the cost function penalize the violation of the balancing constraints.

Note that the OPM problem in (15) is convex but involves $6nH$ optimization variables to be computationally expensive when the BESS is large-scale and has a large number of cells. This issue significantly hinders the applicability of (15) in practical adoptions. Therefore, we will subsequently develop a distributed optimization algorithm where each cell determines its optimal output power with a manageable computational burden.

III. DISTRIBUTED OPM

This section starts with the setup of the distributed OPM as a distributed optimization problem. We then propose the accelerated tracking ADMM algorithm, named as the DOPM algorithm, to address the considered problem.

A. Problem Setup

Let us consider the large-scale BESS shown in Fig. 1 and assume the n modules as independent agents to distributively solve the OPM problem in (15). In sequel, we interchangeably refer to the modules as agents. We can translate the OPM problem in (15) into the following form:

$$\min_{x_j, j=1,...,n} \quad \sum_{j=1}^{n}\sum_{k=1}^{H} f_j(x_j[k]), \tag{16a}$$

$$\text{s.t.} \quad x_j \in X_j, \quad j = 1, ..., n, \tag{16b}$$

$$\sum_{j=1}^{n} Ax_j[k] = b[k], \quad k = 1, ..., H, \tag{16c}$$

where $x_j = \left[P_{b_j} \; P_{l_j} \; E_j \; T_j \; \xi_j^{(E)} \; \xi_j^{(T)} \right]^\top$ collects the optimization variables of module j, $f_j(x_j) = x_j Q x_j^\top$, $Q = \mathbf{diag}(0, 1, 0, 0, \lambda^{(E)}, \lambda^{(T)})$, $A = \mathbf{diag}(1, -1, 0, 0, 0, 0)$, and $b = P_{\text{out}}$. In above, X_j is a feasible set to summarize the safety, balancing, power loss, energy dynamics, and temperature dynamics constraints in (15). The supply-demand balance constraint in (15) is the linear coupling constraint in (16c). We intend to develop a distributed solution to the problem in (16). In the solution, each agent determines its local decision variables x_j to minimize the global cost function in (16a). Each agent must also satisfy the local constraint in (16b), and the global linear coupling constraint in (16c).

Remark 1. *For all $j = 1, ..., n$, the function f_j is strongly convex and the set X_j is convex and compact based on the OPM problem formulation.*

Next, we assume that the modules or agents communicate based on a network topology. The topology graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the node set $\mathcal{V} = \{1, ..., n\}$ and the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represent the agents and the communication links, respectively. The edge $(i, j) \in \mathcal{E}$ if and only if agents i and j communicate. The neighboring agents to agent i are denoted by $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$. It is worth noting that the communication graph does not have to coincide with the electrical connection topology of the cells. We assume $\mathcal{G}$ as undirected and connected. We also assign a weight w_{ij} for the edge (i, j) to indicate agent i's emphasis on information received from agent j. If $(i, j) \notin \mathcal{E}$, $w_{ij} = 0$. We define a consensus matrix $\mathcal{W}$ whose (i, j)th entry is w_{ij}, and impose a balanced information exchange assumption on $\mathcal{W}$, i.e., $\mathcal{W} = \mathcal{W}^\top$ and $\mathcal{W}\mathbb{1}_n = \mathcal{W}^\top \mathbb{1}_n = \mathbb{1}_n$. Note that this assumption is common in the distributed consensus literature [16].

Given the above distributed OPM problem, we propose an accelerated tracking ADMM algorithm to solve (16) so as to overcome the computational complexity facing the centralized OPM problem in (15).

B. Proposed Accelerated Tracking ADMM Algorithm

To begin with, the augmented Lagrangian for (16) is

$$\mathcal{L}(x, \lambda) = \sum_{j=1}^{n}\sum_{k=1}^{H} f_j(x_j[k]) + \lambda^\top d + \frac{\mu}{2}||d||_2^2, \tag{17}$$

where $\lambda \in \mathbb{R}^H$ is the dual variable, and μ is a positive penalty parameter, and

$$\tilde{d}[k] = \sum_{j=1}^{n}(Ax_j[k] - b_j), \tag{18a}$$

$$d = \left[\tilde{d}[1] \quad ... \quad \tilde{d}[H]\right]^\top, \tag{18b}$$

$$b = \sum_{j=1}^{n} b_j. \tag{18c}$$

We can now express the dual problem of (16) as

$$\max_{\lambda} \min_{x \in X} \mathcal{L}(x, \lambda). \tag{19}$$

Here, we assume that the primal problem (16) and the dual problem (19) admit an optimal solution x^* and λ^*, respectively. One can use the ADMM method to solve (19), which alternately updates x and λ in an iterative procedure [17]. The iterations are as follows:

$$x_1^{r+1} = \arg\min_{x_1} \mathcal{L}(x_1, x_2^r, ..., x_n^r; \lambda^r),$$

$$x_j^{r+1} = \arg\min_{x_j} \mathcal{L}(x_1^{r+1}, ..., x_{j-1}^{r+1}, x_j, x_{j+1}^r, ..., x_n^r; \lambda^r),$$

$$\lambda^{r+1} = \lambda^r + \mu d^{r+1},$$

where r represents the iteration number, and d^{r+1} is obtained by applying x_j^{r+1} to (18a) and (18b). Note that d^r measures the infeasibility of the solution x^r. The x-optimization steps of the ADMM algorithm are sequential rather than parallel, slowing down the algorithm. In addition, the λ-update step is not distributed and requires a central node to gather all the decision variables x_j. Some studies have presented parallel ADMM algorithms to address the sequential x-optimization steps for the ADMM [18, 19]. However, these algorithms require a centralized λ-update step.

A tracking ADMM algorithm is developed in [20] to enable a fully distributed solution. We propose to modify the algorithm in this work to accelerate its convergence and improve its computational speed. To fully distribute the update of λ, all agents are assumed to keep a local copy of λ^r and d^r, denoted by λ_j^r and d_j^r, respectively. A consensus

TABLE I: Distributed OPM via Accelerated Tracking ADMM (DOPM)

1: **Initialization**
2: $x_j^0 = \hat{x}_j^0 \in X_j$
3: $d_j^0 = \hat{d}_j^0 = Ax_j^0 - b_j$
4: $\lambda_j^0 = \hat{\lambda}_j^0 \in \mathbb{R}^H$
5: **Repeat until convergence**
6: $\delta_j^r = \sum_{i\in\mathcal{N}_j} w_{ji} d_i^r, \quad \hat{\delta}_j^r = \sum_{i\in\mathcal{N}_j} w_{ji}\hat{d}_i^r$
7: $\hat{l}_j^r = \sum_{i\in\mathcal{N}_j} w_{ji}\hat{\lambda}_i^r$
8: $x_j^{r+1} = \arg\min_{x_j} \sum_{k=1}^{H} f_j(x_j[k]) + \hat{l}_j^{r\top} Ax_{j,1:H} + \frac{\mu}{2}\left\|\hat{\delta}_j^r - A\hat{x}_{j,1:H} + Ax_{j,1:H}\right\|_2^2$
9: $d_j^{r+1} = \delta_j^r + Ax_{j,1:H}^{r+1} - Ax_{j,1:H}^r$
10: $\lambda_j^{r+1} = \hat{l}_j^r + \mu d_j^{r+1}$
11: $\hat{\lambda}_j^{r+1} = \lambda_j^{r+1} + \eta\frac{a_r-1}{a_{r+1}}(\lambda_j^{r+1} - \lambda_j^r)$
12: $\hat{x}_j^{r+1} = x_j^{r+1} + \eta\frac{a_r-1}{a_{r+1}}(x_j^{r+1} - x_j^r)$
13: $\hat{d}_j^{r+1} = \hat{\delta}_j^r + A\hat{x}_{j,1:H}^{r+1} - A\hat{x}_{j,1:H}^r$
14: $r \leftarrow r+1$

scheme can then be applied to enforce the agreement among the local copies as follows:

$$\delta_j^r = \sum_{i\in\mathcal{N}_j} w_{ji} d_i^r, \tag{20a}$$

$$l_j^r = \sum_{i\in\mathcal{N}_j} w_{ji}\lambda_i^r, \tag{20b}$$

where δ_j^r and l_j^r are the local estimates of d^r and λ^r, respectively. The introduction and use of the local variables λ_j^r and d_j^r allow a fully distributed parallel ADMM algorithm. To speed up the convergence, we propose to leverage the Nesterov's acceleration technique in [21]. The Nesterov's technique is based on using previous two optimal points, instead of the mere previous one, for the optimization. We define the accelerated decision variables $\hat{x}_j^{r+1}$ and $\hat{\lambda}_j^{r+1}$ as follows:

$$\hat{x}_j^{r+1} = x_j^{r+1} + \eta\frac{a_r - 1}{a_{r+1}}(x_j^{r+1} - x_j^r), \tag{21a}$$

$$\hat{\lambda}_j^{r+1} = \lambda_j^{r+1} + \eta\frac{a_r - 1}{a_{r+1}}(\lambda_j^{r+1} - \lambda_j^r), \tag{21b}$$

$$a_1 = 1, \quad a_{r+1} = \frac{1+\sqrt{1+4a_r^2}}{2}, \quad r = 1, 2, ... \tag{21c}$$

where $\eta \in (0, 1)$ is the discount factor. We define the infeasibility measure of the $\hat{x}^r$ as follows:

$$\hat{\tilde{d}}^r[k] = \sum_{j=1}^{n}(A\hat{x}_j^r[k] - b_j), \tag{22a}$$

$$\hat{d}^r = \left[\hat{\tilde{d}}^r[1] \quad ... \quad \hat{\tilde{d}}^r[H]\right]^\top. \tag{22b}$$

Note that $\hat{d}^r$ corresponds to d^r in the same way $\hat{x}^r$ does to x^r in (18). We also enforce all the agents to maintain a local copy of $\hat{d}^r$ and agree it that by a consensus scheme, similar to the procedure for d^r in (20a) as follows:

$$\hat{\delta}_j^r = \sum_{i\in\mathcal{N}_j} w_{ji}\hat{d}_i^r. \tag{23}$$

The local copies $\hat{d}_j^r$ will be used in the x-minimization step of the proposed algorithm. After the x-minimization step, we update d_j^r and $\hat{d}_j^r$ as follows:

$$d_j^{r+1} = \delta_j^r + Ax_{j,1:H}^{r+1} - Ax_{j,1:H}^r, \tag{24a}$$

$$\hat{d}_j^{r+1} = \hat{\delta}_j^r + A\hat{x}_{j,1:H}^{r+1} - A\hat{x}_{j,1:H}^r, \tag{24b}$$

where

$$Ax_{j,1:H}^r = \left[Ax_j^r[1] \quad ... \quad Ax_j^r[H]\right]^\top.$$

Given $x_j^0 = \hat{x}_j^0 \in X_j$, $d_j^0 = \hat{d}_j^0 = Ax_j^0 - b_j$, and $\lambda_j^0 = \hat{\lambda}_j^0 \in \mathbb{R}^H$, each agent solves the following optimization problem in parallel:

$$x_j^{r+1} = \arg\min_{x_j} \mathcal{L}(\hat{x}_1^r, \hat{x}_2^r, ..., x_j, ..., \hat{x}_n^r; \hat{\lambda}^r). \tag{25}$$

Expanding the Lagrangian in (25) using (17), one can derive the following:

$$\begin{aligned} x_j^{r+1} = \arg\min_{x_j} &\sum_{j=1}^{n}\sum_{k=1}^{H} f_j(x_j[k]) + \\ &\hat{\lambda}^{r\top}\left(\hat{d}_j^r - A\hat{x}_{j,1:H}^r + Ax_{j,1:H}\right) + \\ &\frac{\mu}{2}\left\|\hat{d}_j^r - A\hat{x}_{j,1:H}^r + Ax_{j,1:H}\right\|_2^2. \end{aligned} \tag{26}$$

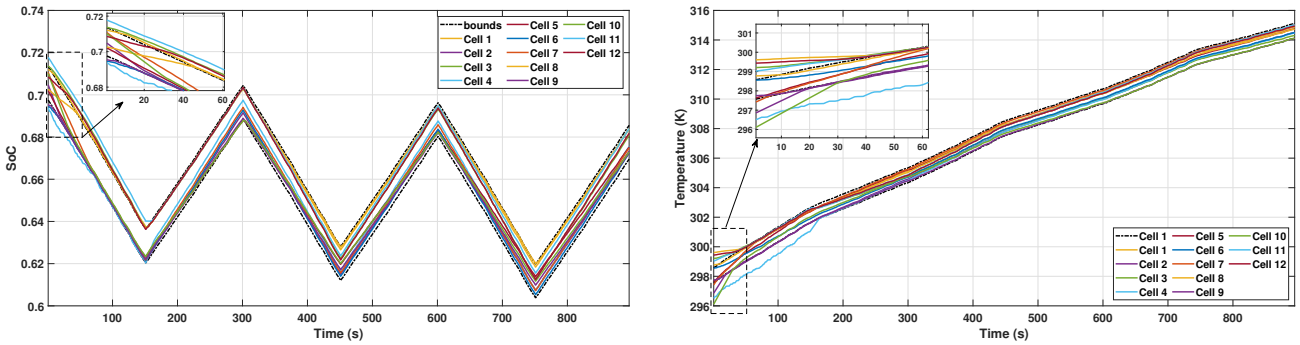
By neglecting the constant terms with respect to x_j, the x-minimization step can be expressed as:

$$\begin{aligned} x_j^{r+1} = \arg\min_{x_j} &\sum_{k=1}^{H} f_j(x_j[k]) + \hat{\lambda}_j^\top Ax_{j,1:H} + \\ &\frac{\mu}{2}\left\|\hat{d}_j^r - A\hat{x}_{j,1:H} + Ax_{j,1:H}\right\|_2^2. \end{aligned} \tag{27}$$

The overall proposed algorithm is summarized in Table I. In the proposed DOPM algorithm, Steps 6 and 7 perform consensus to guarantee the agreement among d_j^r, $\hat{d}_j^r$, and $\hat{\lambda}_j^r$, respectively. Step 8 performs the x-minimizations in parallel among the agents. Step 10 calculates the λ_j^{r+1}, which is fully distributed. The proposed algorithm not only offers fully distributed and parallel optimization but also accelerates the convergence rate, conductive to the OPM of large-scale BESS. The following section will compare the proposed algorithm with the tracking ADMM algorithm for demonstration.

IV. SIMULATION RESULTS

This section presents the simulation results to assess the performance of the proposed DOPM algorithm. The specifications of the considered BESS are summarized in Table II. We consider a receding horizon of ten seconds, i.e., $H = 10$. We also define an arbitrarily connected graph over which the cells communicate. We choose the consensus matrix to be $\mathcal{W} = I - \alpha L$ where I is the identity matrix with appropriate size, L is the Laplacian matrix of the graph, and $\alpha = 0.1$.



(a) (b)

Fig. 2: Simulation results of the SoC and temperature balancing. (a) The SoC of the cells. (b) The temperature of the cells.

TABLE II: Specifications of the BESS

Symbol	Parameter	Value [Unit]
n	Number of battery cells	12
$\bar{Q}$	Cell nominal capacity	2.5 [A.h]
R	Cell internal resistance	31.3 [mΩ]
$[q^{\min}, q^{\max}]$	Cell SoC limits	[0.05,0.95]
$[i^{\min}, i^{\max}]$	Cell current limits	[-7.5,7.5] [A]
C_{th}	Thermal capacitance	40.23 [J/K]
R_{conv}	Convection thermal resistance	41.05 [K/W]
T_{env}	Environment temperature	298 [K]
Δq	SoC balancing threshold	0.8%
ΔT	Temperature balancing threshold	0.5 [K]
Δt	Sampling time	1 [s]

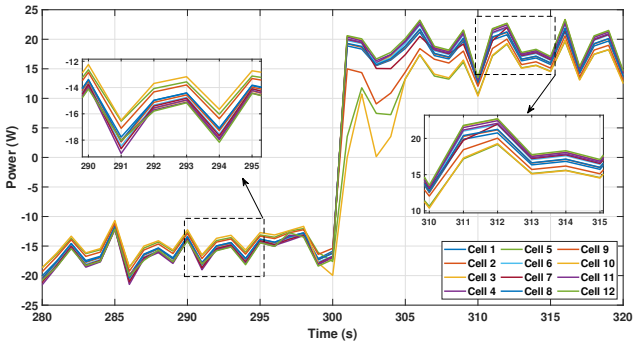


Fig. 3: The output power of the cells.

Note that $\mathcal{W}$ satisfies the balanced information exchange assumption (see [16] for more details). The output power profile for P_{out} is based on periodic charging/discharging, with each cycle lasting for 300 s and using an average output power of 220 W. We use the CVX package to configure and solve the optimization problem [22].

The initial SoC of the cells follows a normal distribution with a mean of 70% and a variance of 1%. The initial temperature of the cells is similarly drawn from a normal distribution with a mean of 298 K and a variance of 1 K.

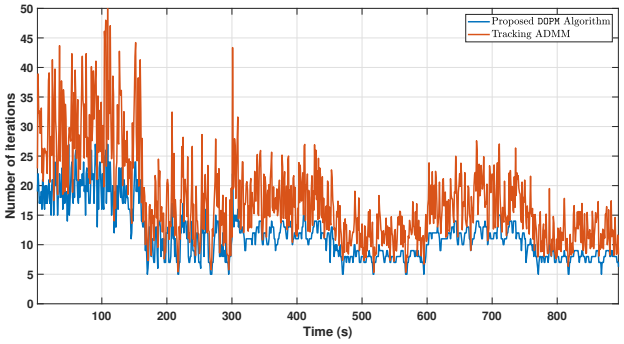


Fig. 4: The number of iterations.

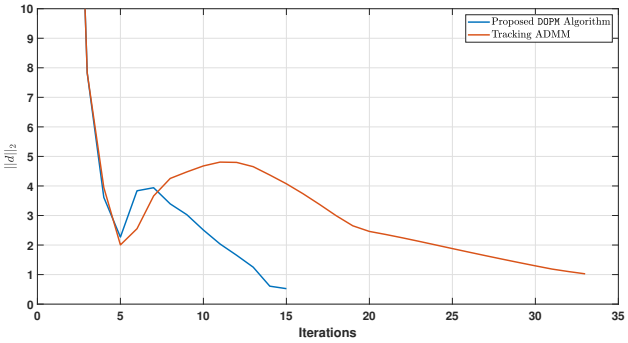


Fig. 5: The evolution of $\|d\|_2$ of the optimization problem at a specific time step.

Further, the cells are made heterogeneous by varying the internal resistances using a zero-mean white Gaussian noise with a variance of 4 mΩ.

Fig. 2 shows the SoC and temperature balancing performance of the proposed DOPM algorithm. According to Fig. 2 (a), the cells' initial SoC are not inside the tolerance bound. However, the proposed DOPM algorithm drives the cells' SoC to reach into the bound after about 160 seconds and continues to optimally regulate the charging and discharging power of the cells to maintain their balanced SoC.

Fig. 2 (b) depicts the cells' temperatures and their deviation from the average, respectively. Initially, the cells' temperatures are not within the desired bound. The proposed

DOPM algorithm successfully controls the cells' temperatures to reach a balanced temperature within about 170 seconds. The cells' temperatures remain balanced afterward, with minor violations due to the slack variables.

To further investigate the proposed DOPM algorithm, Fig. 3 depicts the output power of the cells. For better visualization, we show a magnified view of the time interval of $280 < t < 320$ s. According to Fig. 3, the output power is not equally distributed among the cells. Instead, it is optimally allocated among the cells to minimize the total power losses based on their own conditions and the need for balanced SoC and temperature.

To show the efficacy of the proposed DOPM algorithm, we solve the distributed power management problem with the proposed DOPM and the tracking ADMM algorithm in [20]. Fig. 4 compares the performance of these two algorithms in terms of the number of iterations. Note that the optimization problem generates 720 optimization variables at every time step. We observe that the proposed DOPM algorithm requires fewer iterations to reach the optimal power of the cells. This also implies that the proposed DOPM algorithm needs less computation time for convergence.

Fig. 5 also illustrates the evolution of $\|d\|_2$ over iterations for an arbitrary time step. The optimization algorithms are designed to stop when $\|d\|_2$ falls below 1 W. Note that zero $\|d\|_2$ values indicate that the linear equality constraint is satisfied, i.e., $\sum_{j=1}^{n} Ax_j[k] = b[k],\ k = 1, ..., H$. According to Fig. 5, at this specific time step, the proposed DOPM algorithm stops after 15 iterations, whereas the tracking ADMM algorithm takes 33 iterations.

V. CONCLUSIONS

The sweeping adoption of BESS has stimulated a critical need for OPM to minimize power losses under practical constraints due to safety, cell balancing, and power supply-demand consistency. This paper focuses on enabling distributed OPM that is computationally efficient and scalable to large-scale BESS. We first considered a BESS architecture, which is characterized by converter-based cell-level power control, and formulated a centralized OPM problem based on convex optimization. To substantially reduce the computational cost, we proposed a distributed OPM setup, in which the cells act like independent agents to compute their own decisions toward global OPM while exchanging information with each other. Then, we developed an accelerated tracking ADMM algorithm to address the distributed OPM problem. The algorithm by design combines the tracking ADMM algorithm with an acceleration procedure to deliver a faster computation. Further, the algorithm is fully distributed to require no information fusion or aggregation. The simulation results have corroborated the effectiveness of the proposed algorithm and indicated lower computation time and fewer iterations compared to the literature.

REFERENCES

[1] M. Preindl, C. Danielson, and F. Borrelli, "Performance evaluation of battery balancing hardware," in *European Control Conference*, 2013, pp. 4065–4070.

[2] R. Gu, P. Malysz, M. Preindl, H. Yang, and A. Emadi, "Linear programming based design and analysis of battery pack balancing topologies," in *IEEE Transportation Electrification Conference and Expo (ITEC)*, 2015, pp. 1–7.

[3] J. V. Barreras, C. Pinto, R. de Castro, E. Schaltz, S. J. Andreasen, and R. E. Araújo, "Multi-objective control of balancing systems for li-ion battery packs: A paradigm shift?" in *IEEE Vehicle Power and Propulsion Conference*, 2014, pp. 1–7.

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

[5] 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 Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 11 449–11 461, 2019.

[6] R. de Castro, H. Pereira, R. E. Araújo, J. V. Barreras, and H. C. Pangborn, "qTSL: A multilayer control framework for managing capacity, temperature, stress, and losses in hybrid balancing systems," *IEEE Transactions on Control Systems Technology*, pp. 1–16, 2021.

[7] Q. Ouyang, J. Chen, J. Zheng, and H. Fang, "Optimal multiobjective charging for lithium-ion battery packs: A hierarchical control approach," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4243–4253, 2018.

[8] 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, 2016.

[9] Q. Ouyang, J. Chen, J. Zheng, and H. Fang, "Optimal cell-to-cell balancing topology design for serially connected lithium-ion battery packs," *IEEE Transactions on Sustainable Energy*, vol. 9, no. 1, pp. 350–360, 2018.

[10] T. Yang, D. Wu, H. Fang, W. Ren, H. Wang, Y. Hong, and K. H. Johansson, "Distributed energy resource coordination over time-varying directed communication networks," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 3, pp. 1124–1134, 2019.

[11] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2015.

[12] Y. Kou, Y. Wang, Z. Bie, X. Wang, and T. Ding, "Distributed accelerated descent algorithm for energy resource coordination in multi-agent integrated energy systems," *IET Generation, Transmission & Distribution*, vol. 15, no. 12, pp. 1884–1896, 2021.

[13] A. Farakhor and H. Fang, "A novel modular, reconfigurable battery energy storage system design," in *47th Annual Conference of the IEEE Industrial Electronics Society*, 2021, pp. 1–6.

[14] A. Farakhor, D. Wu, Y. Wang, and H. Fang, "A novel modular, reconfigurable battery energy storage system: Design, control, and experimentation," *IEEE Transactions on Transportation Electrification*, 2023, in press.

[15] N. Murgovski, L. Johannesson, and J. Sjöberg, "Convex modeling of energy buffers in power control applications," *IFAC Proceedings Volumes*, vol. 45, no. 30, pp. 92–99, 2012, 3rd IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling.

[16] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[17] D. Han and X. Yuan, "A note on the alternating direction method of multipliers," *Journal of Optimization Theory and Applications*, vol. 155, no. 1, pp. 227–238, 2012.

[18] D. Bertsekas, P. Tsitsiklis, and N. John, "Parallel and distributed computation: Numeral methods," 1989.

[19] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block ADMM with $\mathcal{O}(1/k)$ convergence," *Journal of Scientific Computing*, vol. 71, no. 2, pp. 712–736, 2017.

[20] A. Falsone, I. Notarnicola, G. Notarstefano, and M. Prandini, "Tracking-ADMM for distributed constraint-coupled optimization," *Automatica*, vol. 117, p. 108962, 2020.

[21] Y. Nesterov, "A method for solving the convex programming problem with convergence rate $\mathcal{O}(1/k^2)$," *Proceedings of the USSR Academy of Sciences*, vol. 269, pp. 543–547, 1983.

[22] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.