Reconfigurable Model Predictive Control for Large Scale Distributed Systems

Jun Chen, *Senior Member, IEEE*

***Abstract*— For large scale systems, centralized control often encounters computational issues. To address this limitation, this paper proposes a new reconfigurable model predictive control (MPC) framework for large scale distributed systems, in which an optimal control problem with time-varying structure is formulated and solved for each control loop. More specifically, at each time step, a subset of the system inputs is dynamically selected to be optimized by MPC, while zero-order-hold is applied to the remaining control inputs. A theoretical upper bound on the performance loss, introduced by the selection of optimization variables, is then derived to guarantee the worst case performance. To minimize the performance loss, this upper bound is in turn used to select the set of inputs to be optimized. The applicability of the proposed approach is illustrated through the battery cell-to-cell balancing problem, where the system has 100 inputs to be optimized in real-time. Numerical results confirm that the proposed approach can achieve better control performance compared to distributed MPC and requires less computation compared to conventional centralized MPC.**

I. Introduction

Control of large distributed systems is of prominent importance for many applications [1]–[3]. Among many approaches, model predictive control (MPC) has been extensively investigated [4]–[9], where distributed MPC can be grouped into non-cooperative distributed MPC, cooperative distributed MPC, and decomposed optimization approach. For example, [3] studies non-cooperative distributed MPC in the context of vehicle platoon. In particular, the system under control is dynamically decoupled and the only coupling is the state constraints and desired states. In other words, each local MPC solves its own optimization problem with local cost function and local terminal constraint formulated using predicted state trajectory from its neighbors' previous prediction. Sufficient condition to guarantee stability is derived and demonstrated through simulation. Reference [8] studies the cooperative distributed MPC for systems that are dynamically coupled, where the terminal set is used to ensure stability. Instead of invariant terminal set, adaptivity is included by formulating it as an optimization problem. The adaptive terminal set avoids over restrictive terminal constraints while guaranteeing stability. Reference [9] studies the distributed MPC without a central coordinator, for interconnected systems through states coupling only. The local predicted state trajectories are communicated to other local controllers, which are then used to formulate optimization problem and constraints.

With the recent advancement of reinforcement learning (RL), multi-agent RL (MARL) has been applied to distributed system, such as distributed process control, traffic signals control, and electrical power networks [10]–[14], where the agents can be fully cooperative, fully competitive, and mixed (neither cooperative nor competitive). For example, [11] studies distributed reinforcement learning problem, where RL algorithm is used to learn local controllers, which is then integrated into the framework of ADMM (alternating direction method of multipliers) for distributed control. In this approach, RL is employed to solve a local optimization problem in a model-free and episodic fashion. Reference [12] studies distributed RL with policy gradient approach, where agreement algorithms is used to efficiently exchange local rewards and experience among agents. Authors in [14] apply advantage actor critic RL algorithm for distributed large scale traffic signal control, where a spatial discount factor is introduced to measure the strength of connection between nearby intersections. Despite recent advancement, the multi-agent reinforcement learning still suffers the scalability issue as its centralized counterpart, as noted in [10].

Compared to distributed control, centralized control has the advantage of achieving optimality and reducing communication among agents, but at the same time requires significant computational power, and hence intractable for large scale systems. To address this issue, this paper proposes a new reconfigurable MPC framework, in which an optimal control problem (OCP) with time-varying structure is formulated and solved for each control loop. In other words, at each time step, a subset of the system inputs is dynamically selected to be optimized by MPC, while zero-order-hold is applied to the remaining control inputs. Such approach reduces the computational requirement of MPC, as the number of optimization variables are significantly reduced. On the other hand, the control performance may be degraded since the control authority is reduced. To quantify this performance loss, a theoretical upper bound is derived to guarantee the worst case performance. Furthermore, to minimize the performance loss, this upper bound is in turn used to select the set of inputs to be optimized. The applicability of the proposed approach is illustrated through the battery cell-to-cell balancing problem, where the system has 100 inputs to be optimized. Numerical results confirm that the proposed approach can achieve better control performance compared to distributed MPC and requires less computation compared to conventional centralized MPC.

This work is supported in part by SECS Faculty Startup Fund and URC Faculty Research Fellowship at Oakland University.

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

The concept of reconfigurable MPC has been studied in literature [15]–[17], where the physical plant is reconfigurable. For example, [16] considers MPC for linear systems with changeable network topology, and proposes a novel reconfiguration control scheme based on ADMM. Reference [17] applies reconfigurable MPC to multievaporator vapor compression systems, where individual evaporators can be turned on or off. A reconfigurable MPC is then designed to accommodate the time varying system configuration. This paper differs from these work as the physical systems components are reconfigurable in [15]–[17], while in our study, the physical systems are assumed to be fixed, but the MPC dynamically selects a subset to form the OCP to reduce the required online computations.

The rest of this paper is organized as follows. Section II provides the problem formulation, while Section III presents the proposed reconfigurable MPC. Theoretical guarantees on performance loss and reconfiguring strategy are discussed in Section IV, with numerical simulation results on cell-to-cell balancing control of 100 connected cells presented in Section V. The paper is concluded in Section VI.

Notations: Throughout the paper, we make use of the following notations and properties. Given a vector v, $\|v\|$ denotes its 2-norm. Furthermore, we denote

$$\|v\|_Q^2 = v^T Q v.$$

Property 1: For a vector v and a symmetric positive semidefinite matrix Q, we have

$$\|v\|_Q^2 = v^T Q v = v^T \left(Q^{1/2}\right)^T Q^{1/2} v = \left\|Q^{1/2} v\right\|^2,$$

where $\|\cdot\|$ without subscript denotes 2-norm.

Property 2: For two vectors v and u, the following inequality holds.

$$\|u+v\|^2 \leq \|u\|^2 + 2\|u\|\|v\| + \|v\|^2 = (\|u\| + \|v\|)^2.$$

II. Problem Formulation

Consider a distributed system with N components, and the nth component has the following dynamics:

$$x_{k+1}^n = A^n x_k^n + B^n u_k^n + B_w^n w_k \tag{1a}$$
$$y_k^n = C^n x_k^n + b^n, \tag{1b}$$

where x^n, u^n and y^n are the states, outputs and inputs for nth component, and w is the disturbance. Denote n_x, n_y and n_u as the number of states, outputs and inputs for each distributed component, respectively. Furthermore, the inputs and outputs of each components are coupled through constraints, as follows:

$$\{u_k^n, n \in \mathcal{N}\} \in \mathcal{U} \subseteq R^{Nn_u} \tag{2a}$$
$$\{y_k^n, n \in \mathcal{N}\} \in \mathcal{Y} \subseteq R^{Nn_y}, \tag{2b}$$

where $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of all distributed systems.

Remark 1: Note that the distributed systems (1)-(2) are coupled by the disturbance w_k as well as the input and output constraints. Furthermore, though we consider n_x, n_y and n_u are the same for all components, the proposed work can be straightforwardly extended to include case where each component can have different dimensions.

At each time step, MPC solves the following optimal control problem (OCP) over a prediction horizon p:

$$\min_{u_k^n} \quad J = \sum_{n=1}^{N}\sum_{k=1}^{p} ||y_k^n||_{Q_y}^2 + \sum_{n=1}^{N}\sum_{k=0}^{p-1} ||u_k^n||_{Q_u}^2 \tag{3a}$$
$$\text{s.t.} \quad \text{System dynamics } (1), \qquad \forall n \tag{3b}$$
$$x_k^n = \tilde{x}_k^n, \qquad \forall n \tag{3c}$$
$$\text{input and output constraints } (2), \qquad \forall k. \tag{3d}$$

Note that the weight matrices Q_y and Q_u are assumed to be symmetric positive semidefinite. It is then trivial to see that the total number of optimization variables is Npn_u. When N is large, solving the above OCP is intractable (even for small p) due to the curse of dimensionality. To address this issue, in this paper, a reconfigurable MPC is proposed where a subset of components is dynamically selected to form the OCP, while for the remaining components, zero-order-hold is applied to their previous control commands. In the sequel, we will discuss in detail the formal formulation of OCP with only a subset of components and its performance loss.

III. Reconfigurable MPC

Given a subset $\mathcal{W} \subseteq \mathcal{N}$, denote its complementary set as $\overline{\mathcal{W}} = \mathcal{N} - \mathcal{W}$. This section presents the formulation of a reduced size OCP that only includes components in $\mathcal{W}$. We first make the following definitions and assumptions.

Definition 1: For the nth component, given its input u^n, define $x^n(u^n)$ as the state sequence that is obtained by integrating (1) using $\bar{u}^n$. Further define $y^n(\bar{u}^n) = C^n x^n(\bar{u}^n) + b^n$ as the corresponding output sequence.

Assumption 1: For components $n \in \overline{\mathcal{W}}$, a control input $\bar{u}^n$ is available.

Definition 2: Given $\mathcal{W}$, $\overline{\mathcal{W}} = \mathcal{N} - \mathcal{W}$, and $\bar{u}^n$ and $y^n(\bar{u}^n)$ for each $n \in \overline{\mathcal{W}}$, define the set of feasible input for $n \in \mathcal{W}$ as

$$\hat{\mathcal{U}} = \{\{u^n, n \in \mathcal{W}\} | \{u^n, n \in \mathcal{N}\} \in \mathcal{U}\} \subseteq R^{|\mathcal{W}|n_u}, \tag{4}$$

and the set of feasible output for $n \in \mathcal{W}$ as

$$\hat{\mathcal{Y}} = \{\{y^n(u^n), n \in \mathcal{W}, u^n \in \hat{\mathcal{U}}\} \\ | \{y^n, n \in \mathcal{N}\} \in \mathcal{Y}\} \subseteq R^{|\mathcal{W}|n_y}. \tag{5}$$

Now we are ready to formulate a reduced size OCP that only includes components in $\mathcal{W}$, as follows.

$$\min_{u_k^n, n \in \mathcal{W}} \quad J = \sum_{n \in \mathcal{W}}\sum_{k=1}^{p} ||y_k^n||_{Q_y}^2 + \sum_{n \in \mathcal{W}}\sum_{k=0}^{p-1} ||u_k^n||_{Q_u}^2 \tag{6a}$$
$$\text{s.t.} \quad \text{System constraints } (1), \qquad \forall n \in \mathcal{W} \tag{6b}$$
$$x_k^n = \tilde{x}_k^n, \qquad \forall n \in \mathcal{W} \tag{6c}$$
$$\{u_k^n, n \in \mathcal{W}\} \in \hat{\mathcal{U}}, \qquad \forall k \tag{6d}$$
$$\{y_k^n, n \in \mathcal{N}\} \in \hat{\mathcal{Y}}, \qquad \forall k. \tag{6e}$$

For each control loop, the proposed reconfigurable MPC selects $\mathcal{W}$, solves OCP (6), and assembles the control vector $u(\mathcal{W}) = \{\hat{u}_k^n\}$ according to the following.

$$\hat{u}_k^n = \begin{cases} \text{solution of (6)} & \text{if } n \in \mathcal{W} \\ \bar{u}^n & \text{if } n \in \overline{\mathcal{W}}. \end{cases} \tag{7}$$

Therefore, we denote the MPC with subset $\mathcal{W}$ as MPC($\mathcal{W}$). It is then trivial to see that MPC that solves the full size OCP (3) is equivalent to MPC($\mathcal{N}$).

Remark 2: Note that MPC($\mathcal{W}$) only optimizes inputs u^n for components in $\mathcal{W}$, while for $n \in \overline{\mathcal{W}}$, its control input $\bar{u}^n$ is assumed to be available according to Assumption 1, which is also used to form the constraints in (6d) and (6e). Therefore, the number of optimization variables of MPC($\mathcal{W}$) is reduced to $|\mathcal{W}|pn_u$.

Remark 3: Assumption 1 guarantees the existence of $\bar{u}^n$ for $n \in \overline{\mathcal{W}}$, Note that Assumption 1 is not restrictive since one can always apply zero-order-hold to previous control command for $n \in \overline{\mathcal{W}}$.

To ensure that OCP (6) is feasible, we make the following assumption.

Assumption 2: Given $\mathcal{W}$, $\overline{\mathcal{W}}$, and $\bar{u}^n$ and $y^n(\bar{u}^n)$ for each $n \in \overline{\mathcal{W}}$, we assume $\hat{\mathcal{U}} \neq \emptyset$ and $\hat{\mathcal{Y}} \neq \emptyset$.

Now that we have formulated MPC($\mathcal{W}$) as in (6), we need to address the following questions.

(Q1) What is the performance loss by solving MPC($\mathcal{W}$) instead of MPC($\mathcal{N}$)?

(Q2) How to select $\mathcal{W}$ in real-time to minimize the performance loss?

IV. Performance Loss and Subsection Selection

A. Performance Loss

To answer Q1 above, we start by assuming $\mathcal{W}$ is selected, and provide an upper bound on the performance loss. Given an input sequence $u = \{u_k^n\}$, $n \in \mathcal{N}$, with a slight abuse of notation, define the performance index as,

$$J(u) = \sum_{n=1}^{N}\sum_{k=1}^{p} ||y_k^n(u_k^n)||_{Q_y}^2 + \sum_{n=1}^{N}\sum_{k=0}^{p-1} ||u_k^n||_{Q_u}^2. \tag{8}$$

Then the performance loss due to optimizing $\mathcal{W}$ can be represented by

$$L(\mathcal{W}) = J(u(\mathcal{W})) - J(u(\mathcal{N})). \tag{9}$$

The next lemma provides an upper bound for $J(u(\mathcal{N}))$.

Lemma 1: Given system (1) and performance index (8), $J(u(\mathcal{N}))$ is upperbounded by $Np\left\|Q_y\right\|\left\|\Delta_y\right\|^2 + Np\left\|Q_u\right\|\left\|\Delta_u\right\|^2$, where

$$\Delta_u = \max_{u\in\mathcal{U}} \|u\|, \qquad \Delta_y = \max_{y\in\mathcal{Y}} \|y\|. \tag{10}$$

Proof:

$$\begin{aligned} J(u(\mathcal{N})) =& \sum_{n=1}^{N}\sum_{k=1}^{p} \|y_k^n\|_{Q_y}^2 + \sum_{n=1}^{N}\sum_{k=0}^{p-1} \|u_k^n\|_{Q_u}^2 \\ =& \sum_{n=1}^{N}\sum_{k=1}^{p} \left\|Q_y^{1/2} y_k^n\right\|^2 + \sum_{n=1}^{N}\sum_{k=0}^{p-1} \left\|Q_u^{1/2} u_k^n\right\|^2 \\ \leq& \sum_{n=1}^{N}\sum_{k=1}^{p} \|Q_y\| \, \|y_k^n\|^2 + \sum_{n=1}^{N}\sum_{k=0}^{p-1} \|Q_u\| \, \|u_k^n\|^2 \\ \leq& \|Q_y\| \sum_{n=1}^{N}\sum_{k=1}^{p} \|\Delta_y\|^2 \\ &+ \|Q_u\| \sum_{n=1}^{N}\sum_{k=0}^{p-1} \|\Delta_u\|^2 \\ =& Np\|Q_y\| \, \|\Delta_y\|^2 + Np\|Q_u\| \, \|\Delta_u\|^2. \end{aligned} \tag{11}$$

This completes the proof. ■

To derive an upperbound for $L(\mathcal{W})$, we first make the following definition.

Definition 3: Given $\mathcal{W}$, denote $u(\mathcal{N}) = \{u_k^n\}$ and $u(\mathcal{W}) = \{\hat{u}_k^n\} = \{u_k^n + \delta_{u,k}^n\}$. Define the maximum difference between u_k^n and $\hat{u}_k^n$ for all k and n as δ_u, i.e.,

$$\delta_u = \max_{k,n} \left\|\delta_{u,k}^n\right\| = \max_{k,n} \|u_k^n - \hat{u}_k^n\|. \tag{12}$$

Given A^n, B^n, and C^n as in (1), define

$$M_k^n = \sum_{i=1}^{k} \left\| C^n \left(A^n\right)^{i-1} B^n \right\|. \tag{13}$$

Then the following theorem provides an upperbound for $L = J(u(\mathcal{W})) - J(u(\mathcal{N}))$.

Theorem 1: Given $\mathcal{W}$, the performance loss L of MPC($\mathcal{W}$) compared to MPC($\mathcal{N}$) is upperbounded by

$$\begin{aligned} L(\mathcal{W}) \leq& 3pN\delta_u^2 \|Q_u\| + 2\delta_u \|Q_u\| \sum_{n=1}^{N}\sum_{k=0}^{p-1} (||\hat{u}_k^n||) \\ &+ 3\delta_u^2 \|Q_y\| \sum_{n=1}^{N}\sum_{k=1}^{p} (M_k^n)^2 \\ &+ 2\delta_u \|Q_y\| \sum_{n=1}^{N}\sum_{k=1}^{p} (M_k^n \|C^n \hat{x}_k^n + b^n\|). \end{aligned} \tag{14}$$

The proof is provided in Appendix.

Remark 4: Note that comparing the upperbound (14) of the performance loss $L(\mathcal{W})$ with the upperbound (11) of the performance index $J(u(\mathcal{N}))$, it is apparent that (14) is useful only when δ_u is sufficiently small. Otherwise (14) can become overly conservative.

B. Selection of $\mathcal{W}$

To address Q2, the goal is to develop a mechanism for selecting $\mathcal{W}$ in real-time to minimize the performance loss $L(\mathcal{W})$. In other words, at each time step, we want to find $\mathcal{W}$ such that

$$\mathcal{W} = \arg\min_{\mathcal{W}} L(\mathcal{W}). \tag{15}$$

The next two lemmas states that $\mathcal{N}$ is the solution to (15).

Lemma 2: Given $\mathcal{W}_1$ and $\mathcal{W}_2$ such that $\mathcal{W}_1 \subset \mathcal{W}_2$, then $L(\mathcal{W}_1) \geq L(\mathcal{W}_2)$.

Proof: Denote the solution of MPC($\mathcal{W}_1$) as $\hat{u}_{1,k}^n$, and the solution of MPC($\mathcal{W}_2$) as $\hat{u}_{2,k}^n$. Then we have

$$J(u(\mathcal{W}_2)) = J(\hat{u}_{2,k}^n)$$

$$
\begin{aligned}
&=\sum_{n=1}^{N}\sum_{k=1}^{p}||y_k^n(\hat{u}_{2,k}^n)||_{Q_y}^2+\sum_{n=1}^{N}\sum_{k=0}^{p-1}||\hat{u}_{2,k}^n||_{Q_u}^2\\
&=\sum_{n\in\mathcal{W}_1}\sum_{k=1}^{p}||y_k^n(\hat{u}_{2,k}^n)||_{Q_y}^2+\sum_{n\in\mathcal{W}_1}\sum_{k=0}^{p-1}||\hat{u}_{2,k}^n||_{Q_u}^2\\
&\quad+\sum_{n\in\mathcal{W}_2-\mathcal{W}_1}\sum_{k=1}^{p}||y_k^n(\hat{u}_{2,k}^n)||_{Q_y}^2\\
&\quad+\sum_{n\in\mathcal{W}_2-\mathcal{W}_1}\sum_{k=0}^{p-1}||\hat{u}_{2,k}^n||_{Q_u}^2\\
&\quad+\sum_{n\notin\mathcal{W}_2}\sum_{k=1}^{p}||y_k^n(\hat{u}_{2,k}^n)||_{Q_y}^2\\
&\quad+\sum_{n\notin\mathcal{W}_2}\sum_{k=0}^{p-1}||\hat{u}_{2,k}^n||_{Q_u}^2\\
&=\sum_{n\in\mathcal{W}_1}\sum_{k=1}^{p}||y_k^n(\hat{u}_{2,k}^n)||_{Q_y}^2+\sum_{n\in\mathcal{W}_1}\sum_{k=0}^{p-1}||\hat{u}_{2,k}^n||_{Q_u}^2\\
&\quad+\sum_{n\in\mathcal{W}_2-\mathcal{W}_1}\sum_{k=1}^{p}||y_k^n(\hat{u}_{2,k}^n)||_{Q_y}^2\\
&\quad+\sum_{n\in\mathcal{W}_2-\mathcal{W}_1}\sum_{k=0}^{p-1}||\hat{u}_{2,k}^n||_{Q_u}^2\\
&\quad+\sum_{n\notin\mathcal{W}_2}\sum_{k=1}^{p}||y_k^n(u^n)||_{Q_y}^2+\sum_{n\notin\mathcal{W}_2}\sum_{k=0}^{p-1}||u^n||_{Q_u}^2\\
&\leq\sum_{n\in\mathcal{W}_1}\sum_{k=1}^{p}||y_k^n(\hat{u}_{1,k}^n)||_{Q_y}^2+\sum_{n\in\mathcal{W}_1}\sum_{k=0}^{p-1}||\hat{u}_{1,k}^n||_{Q_u}^2\\
&\quad+\sum_{n\in\mathcal{W}_2-\mathcal{W}_1}\sum_{k=1}^{p}||y_k^n(u^n)||_{Q_y}^2\\
&\quad+\sum_{n\in\mathcal{W}_2-\mathcal{W}_1}\sum_{k=0}^{p-1}||u^n||_{Q_u}^2\\
&\quad+\sum_{n\notin\mathcal{W}_2}\sum_{k=1}^{p}||y_k^n(u^n)||_{Q_y}^2+\sum_{n\notin\mathcal{W}_2}\sum_{k=0}^{p-1}||u^n||_{Q_u}^2\\
&=J(\hat{u}_{1,k}^n)=J(u(\mathcal{W}_1)).
\end{aligned}
\tag{16}
$$

Now we have

$$
\begin{aligned}
L(\mathcal{W}_1)=&J(u(\mathcal{W}_1))-J(u(\mathcal{N}))\\
\geq&J(u(\mathcal{W}_2))-J(u(\mathcal{N})))\\
=&L(\mathcal{W}_2).
\end{aligned}
\tag{17}
$$

This completes the proof. ■

Lemma 3: Given two positive integers $d_1 < d_2 \leq N$, let

$$
\begin{aligned}
\mathcal{W}_1&=\arg\min_{\mathcal{W},|\mathcal{W}|=d_1}L(\mathcal{W})\\
\mathcal{W}_2&=\arg\min_{\mathcal{W},|\mathcal{W}|=d_2}L(\mathcal{W}).
\end{aligned}
$$

Then we have

$$
L(\mathcal{W}_1)\geq L(\mathcal{W}_2). \tag{18}
$$

Proof: Let $\mathcal{W}_1=\arg\min_{\mathcal{W},|\mathcal{W}|=d_1}L(\mathcal{W})$. Select any $\mathcal{W}_3\supset\mathcal{W}_1$ such that $|\mathcal{W}_3|=d_2$. According to Lemma 2, we have

$$
L(\mathcal{W}_1)\geq L(\mathcal{W}_3).
$$

By the definition of $\mathcal{W}_2$, we have

$$
L(\mathcal{W}_2)=\min_{\mathcal{W},|\mathcal{W}|=d_2}L(\mathcal{W})\leq L(\mathcal{W}_3).
$$

Putting both inequalities together, we have

$$
L(\mathcal{W}_1)\geq L(\mathcal{W}_3)\geq L(\mathcal{W}_2).
$$

This completes the proof. ■

According to Lemma 3, it is obvious that

$$
\arg\min_{\mathcal{W}}L(\mathcal{W})=\arg\min_{\mathcal{W},|\mathcal{W}|=N}L(\mathcal{W})=\mathcal{N}, \tag{19}
$$

i.e., $\mathcal{N}$ is the solution to (15), making (15) a trivial criteria for $\mathcal{W}$ selection. Note that setting $\mathcal{W}=\mathcal{N}$ is equivalent to solving the full size MPC, and hence providing no computational benefit. Therefore, at each time step, instead of selecting $\mathcal{W}$ from the power set of $\mathcal{N}$, we fixed the size of $\mathcal{W}$ by a pre-selected integer d. In other words,

$$
\mathcal{W}=\arg\min_{\mathcal{W},|\mathcal{W}|=d}L(\mathcal{W}), \tag{20}
$$

where d is selected to balance control performance and computation. Note that we want to select $\mathcal{W}$ prior to solving any OCP. However, utilizing (20) to select $\mathcal{W}$ requires solving MPC($\mathcal{W}$) for all potential $\mathcal{W}$ such that $|\mathcal{W}|=d$. Fortunately, Theorem 1 provides some useful information regarding $\mathcal{W}$ that does not require solving any optimization problem. Define

$$
\begin{aligned}
L_{\mathcal{W}}=&2\delta_u\left\|Q_u\right\|\sum_{n\in\mathcal{W}}\sum_{k=0}^{p-1}(||\hat{u}_k^n||)\\
&+3\delta_u^2\left\|Q_y\right\|\sum_{n\in\mathcal{W}}\sum_{k=1}^{p}(M_k^n)^2\\
&+2\delta_u\left\|Q_y\right\|\sum_{n\in\mathcal{W}}\sum_{k=1}^{p}(M_k^n\left\|C^n\hat{x}_k^n+b^n\right\|)
\end{aligned}
\tag{21}
$$

$$
\begin{aligned}
L_{\overline{\mathcal{W}}}=&2\delta_u\left\|Q_u\right\|\sum_{n\in\overline{\mathcal{W}}}\sum_{k=0}^{p-1}(||\bar{u}^n||)\\
&+3\delta_u^2\left\|Q_y\right\|\sum_{n\in\overline{\mathcal{W}}}\sum_{k=1}^{p}(M_k^n)^2\\
&+2\delta_u\left\|Q_y\right\|\sum_{n\in\overline{\mathcal{W}}}\sum_{k=1}^{p}(M_k^n\left\|C^n\bar{x}_k^n+b^n\right\|).
\end{aligned}
\tag{22}
$$

Then the upper bound given in (14) on performance loss $L(\mathcal{W})$ can be equivalently represented as

$$
L(\mathcal{W})\leq 3pN\delta_u^2\left\|Q_u\right\|+L_{\mathcal{W}}+L_{\overline{\mathcal{W}}}. \tag{23}
$$

Here the first term of (23) is constant regardless of the selection of $\mathcal{W}$, while the second and third terms are dependent on the selection of $\mathcal{W}$. Furthermore, the computation of the second term $L_{\mathcal{W}}$ requires solving MPC($\mathcal{W}$), while the third

Algorithm 1: Reconfigurable Model Predictive Control

```
Input: M_k^n, t, ū^n, d, x̃^n, dynamics (1)
Output: u, ū^n
if t = 0 then
    W ← N;
else
    for n = 1 to N do
        {x̄_k^n} ← Integrating (13) using ū^n and x̃^n;
        L_{W̄,n} ←(26);
    end
    W ← (27);
end
Solve MPC(W) as formulated by (6);
for n ∈ W do
    û^n ← Solution of (6);
    ū^n ← û^n;
end
for n ∈ W̄ do
    û^n ← ū^n;
end
u ← {û_0^n}
```

term $L_{\overline{\mathcal{W}}}$ can be computed prior to solving any optimization problem. Therefore, in this paper, we use $L_{\overline{\mathcal{W}}}$ to select $\mathcal{W}$. More specifically, at each time step, given d, $\mathcal{W}$ is selected as

$$\mathcal{W} = \arg \min_{\mathcal{W}, |\mathcal{W}|=d} L_{\overline{\mathcal{W}}}. \tag{24}$$

Remark 5: Equ. (20) selects $\mathcal{W}$ that minimizes the control performance loss. However, this would require solving MPC($\mathcal{W}$) prior to its selection. The idea of (24) is then to select $\mathcal{W}$ such that the upper bound of performance loss is minimized. Numerical analysis will be presented in the sequel to demonstrate that such compromise does not impact the overall control performance.

To solve (24), we first rewrite $L_{\overline{\mathcal{W}}}$ as

$$L_{\overline{\mathcal{W}}} = \sum_{n \in \overline{\mathcal{W}}} L_{\overline{\mathcal{W}},n} \tag{25}$$

$$\begin{aligned} L_{\overline{\mathcal{W}},n} =& 2\delta_u \|Q_u\| \sum_{k=0}^{p-1} (||\bar{u}^n||) + 3\delta_u^2 \|Q_y\| \sum_{k=1}^{p} (M_k^n)^2 \\ &+ 2\delta_u \|Q_y\| \sum_{k=1}^{p} (M_k^n \|C^n \bar{x}_k^n + b^n\|) . \end{aligned} \tag{26}$$

Given $\{L_{\overline{\mathcal{W}},n}\}$, define the set of d large elements of $\{L_{\overline{\mathcal{W}},n}\}$ as $\texttt{max\_d}(\{L_{\overline{\mathcal{W}},n}\})$. Utilizing the fact that $L_{\overline{\mathcal{W}},n}$ for each n is independent of each other, we have

$$\mathcal{W} = \left\{ n \in \mathcal{W} | L_{\overline{\mathcal{W}},n} \in \texttt{max\_d}(\{L_{\overline{\mathcal{W}},n}\}) \right\} . \tag{27}$$

Putting everything together, the proposed reconfigurable MPC, denoted as ReMPC for short, is summarized in Algorithm 1.

V. Battery Cell Balancing Control

To illustrate the applicability of the proposed ReMPC, consider the battery cell balancing system in [18], which consists of N battery cells connected in series. The system dynamics of cell n can be modeled as [19]:

$$s_{k+1}^n = s_k^n - \frac{I_k + u_k^n}{3600 C^n} T_s \tag{28a}$$

$$v_{k+1}^n = v_k^n - \frac{T_s}{R_p^n C_p^n} v_k^n + \frac{I_k + u_k^n}{C_p^n} T_s \tag{28b}$$

$$v_{o,k}^n = V_{oc,k}^n - v_{k+1}^n - (I_k + u_k^n) R_o^n, \tag{28c}$$

where s^n is the state-of-charge (SOC) of cell n, v^n is the relaxation voltage, v_o^n is the terminal voltage, C^n is the cell capacity, R_p^n is the relaxation resistance, C_p^n is the relaxation capacitor, V_{oc}^n is the open circuit voltage and is SOC dependent, R_o is the output resistance, I_k is the current of the string, and u_k^n is the balancing current.

Due to cell variation and degradation, the cell capacity of C^n of each cell can be different, resulting in different SOC levels among cells. During battery operation, whenever one cell's SOC falls below 0 or its terminal voltage v_o^n falls below certain lower bound, the whole battery operation is halted due to safety reason, though by that time there can be other cells with higher SOC. To alleviate this issue, cell balancing control has been studied in literature [18], [20]–[23] to actively transport charge from cell to cell, through balancing current u^n, to maintain a balanced SOC and terminal voltages among cells. The aforementioned work has shown promising potential of extending battery operating range through active balancing. However, due to high computational load, existing study often simulates a battery string with a few cells. In this work, we apply the proposed ReMPC discussed and analyzed in previous sections to active cell balancing control problem, considering a large number of connected cells, e.g. $N = 100$.

For time step k, let $\bar{s}_k$ be the balanced SOC level that we want all cells to track, then the output of the prediction model can be written as

$$y_k^n = \begin{bmatrix} s_k^n \\ v_{o,k}^n \end{bmatrix} - \begin{bmatrix} \bar{s}_k \\ 0 \end{bmatrix} . \tag{28d}$$

The constraints are defined as

$$\mathcal{U} = \left\{ u_k^n \mid \sum_n u_k^n = 0 \right\} \tag{29}$$

$$\mathcal{Y} = \left\{ y_k^n \mid s_k^n \geq 0, v_{o,k}^n \geq v_{\text{min}} \right\} , \tag{30}$$

where v_{min} is the minimum voltage bound below which the battery operation is halted.

Three variants of the proposed ReMPC, i.e., ReMPC with $d = 10$, $d = 40$, and $d = 70$, are implemented and denoted as ReMPC(10), ReMPC(40), and ReMPC(70), respectively. Fig. 1 plots the normalized performance loss compared to a centralized MPC (denoted as CMPC). As can be seen, despite the large percentage of performance loss in the beginning, the performance loss is dropped below 10% quickly. Furthermore, as d increases, the performance

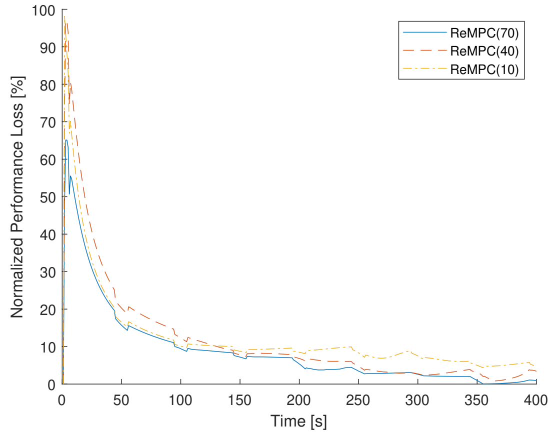

Fig. 1. Normalized performance loss L for different d.

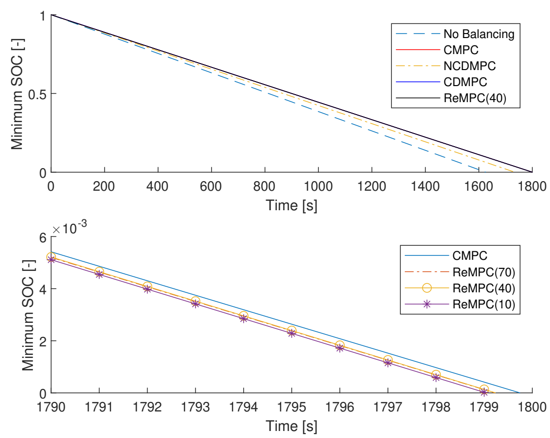

Fig. 2. *Top*: Comparison of minimum SOC for different controllers. *Bottom*: Comparison of minimum SOC for ReMPC with different d.

loss decreases as well. In particular, ReMPC(70) can achieve almost 0% performance loss, as time increases.

To further compare the effectiveness of the proposed ReMPC, we also implement non-cooperative distributed MPC and cooperative distributed MPC as presented in [4], which are denoted as NCDMPC and CDMPC, respectively. Fig. 2 compares the minimum SOC among all cells for different controllers. As can be seen, without balancing control, the minimum SOC drops below 0 at time 1,626 seconds, while with NCDMPC the battery operation is extended to 1,735 seconds. Finally, CMPC, CDMPC, ReMPC(10), ReMPC(40) and ReMPC(70) can all achieve 1,799 seconds of battery operation. Note that from Fig. 2 it appears that CMPC can achieve longer battery operation compared to ReMPC. However, the extension is less than the control sampling time $T_s = 1$, and hence it is ignored in the subsequent discussion.

Finally, Table I summarizes the battery balancing results for all controllers, together with the relative computation compared to CMPC. It is worth noting that the proposed ReMPC does not incur any major control performance degradation, while at the same time reduced significant amount of computation compred to CMPC. In particular, ReMPC(10)

TABLE I
COMPARISON OF BATTERY BALANCING RESULTS

Controller	Range [s]	Extension	Relative Computation
No balancing	1,626	-	-
CMPC	1,799	10.701%	100%
$d = 10$	1,799	10.701%	4.77%
$d = 40$	1,799	10.701%	20.00%
$d = 70$	1,799	10.701%	53.57%
NCDMPC	1,735	6.704%	24.95%
CDMPC	1,799	10.701%	48.25%

requires only 4.77% of relative computation, ReMPC(40) requires 20%, while ReMPC(50) requires 53.57%. Compared to distributed MPC approach, ReMPC can achieve better control performance compared to NCDMPC, while requires less computation compared to CDMPC.

VI. CONCLUSION

This paper proposes a new reconfigurable model predictive control (MPC) framework for large scale distributed systems, in which an optimal control problem (OCP) with time-varying structure is formulated and solved for each control loop. More specifically, at each time step, a subset of the system inputs is dynamically selected to be optimized by MPC, while zero-order-hold is applied to the remaining inputs. A theoretical upper bound on performance loss is then derived to guarantee the worst case performance, which is later used to select the set of inputs to be optimized. Advantage over conventional centralized MPC and distributed MPC is clearly demonstrated through numerical example on the battery cell-to-cell balancing control with 100 cells. Future work includes the calculation of d to explicitly balance performance loss and computation.

APPENDIX

This appendix provides proof of Theorem 1.

Proof: Denote the first term of (8) as J_u. Then we have

$$
\begin{aligned}
L_u =& J_u(u(\mathcal{W})) - J_u(u(\mathcal{N})) \\
=& \sum_{n=1}^{N}\sum_{k=0}^{p-1}\left(\left\|Q_u^{1/2}u_k^n + Q_u^{1/2}\delta_{u,k}^n\right\|^2 - \left\|Q_u^{1/2}u_k^n\right\|^2\right) \\
\leq& \sum_{n=1}^{N}\sum_{k=0}^{p-1}\left(\left\|Q_u^{1/2}u_k^n\right\|^2 + \left\|Q_u^{1/2}\delta_{u,k}^n\right\|^2\right. \\
&\left. + 2\left\|Q_u^{1/2}u_k^n\right\|\left\|Q_u^{1/2}\delta_{u,k}^n\right\| - \left\|Q_u^{1/2}u_k^n\right\|^2\right) \\
=& \sum_{n=1}^{N}\sum_{k=0}^{p-1}\left(\left\|Q_u^{1/2}\delta_{u,k}^n\right\|^2\right. \\
&\left. + 2\left\|Q_u^{1/2}u_k^n\right\|\left\|Q_u^{1/2}\delta_{u,k}^n\right\|\right) \\
=& \sum_{n=1}^{N}\sum_{k=0}^{p-1}\left(\left\|Q_u^{1/2}\delta_{u,k}^n\right\|^2\right.
\end{aligned}
$$

$$+2\left\|Q_u^{1/2}\hat{u}_k^n - Q_u^{1/2}\delta_{u,k}^n\right\|\left\|Q_u^{1/2}\delta_{u,k}^n\right\|\Big)$$
$$\leq \sum_{n=1}^{N}\sum_{k=0}^{p-1}\Bigg(\left\|Q_u^{1/2}\delta_{u,k}^n\right\|^2 + 2\left[\left\|Q_u^{1/2}\hat{u}_k^n\right\| + \left\|Q_u^{1/2}\delta_{u,k}^n\right\|\right]\times\left\|Q_u^{1/2}\delta_{u,k}^n\right\|\Bigg)$$
$$=\sum_{n=1}^{N}\sum_{k=0}^{p-1}\Bigg(3\left\|Q_u^{1/2}\delta_{u,k}^n\right\|^2 + 2\left\|Q_u^{1/2}\hat{u}_k^n\right\|\left\|Q_u^{1/2}\delta_{u,k}^n\right\|\Bigg)$$
$$\leq \sum_{n=1}^{N}\sum_{k=0}^{p-1}\left(3\left\|Q_u\right\|\left\|\delta_{u,k}^n\right\|^2 + 2\left\|Q_u\right\|\left\|\hat{u}_k^n\right\|\left\|\delta_{u,k}^n\right\|\right)$$
$$\leq \sum_{n=1}^{N}\sum_{k=0}^{p-1}\left(3\delta_u^2\left\|Q_u\right\| + 2\delta_u\left\|Q_u\right\|\left\|\hat{u}_k^n\right\|\right)$$
$$=3pN\delta_u^2\left\|Q_u\right\| + 2\delta_u\left\|Q_u\right\|\sum_{n=1}^{N}\sum_{k=0}^{p-1}\left(||\hat{u}_k^n||\right). \quad (31)$$

Next, to derive a relationship between predictive state x_k^n and the control sequence u_0^n, u_1^n, $\cdots$, u_{k-1}^n, we have,

$$\begin{aligned}
x_k^n =& A^n x_{k-1}^n + B^n u_{k-1}^n + B_w^n w_{k-1}\\
=& A^n(A^n x_{k-2}^n + B^n u_{k-2}^n + B_w^n w_{k-2})\\
&+ B^n u_{k-1}^n + B_w^n w_{k-1}\\
=& \left(A^n\right)^2 x_{k-2}^n + A^n B^n u_{k-2}^n + B^n u_{k-1}^n\\
&+ A^n B_w^n w_{k-2} + B_w^n w_{k-1}\\
&\vdots\\
=& \left(A^n\right)^k x_0^n + \sum_{i=1}^{k}\left(\left(A^n\right)^{i-1}B^n u_{k-i}^n\right)\\
&+ \sum_{i=1}^{k}\left(\left(A^n\right)^{i-1}B_w^n w_{k-i}\right).
\end{aligned}$$

Denote the state sequence corresponding to $\mathcal{N}$ as $x(\mathcal{N}) = \{x_k^n\}$, and $x(\mathcal{W}) = \{\hat{x}_k^n\} = \{x_k^n + \delta_{x,k}^n\}$. Then we have

$$\begin{aligned}
\delta_{x,k}^n =& \hat{x}_k^n - x_k^n\\
=& \left(A^n\right)^k x_0^n + \sum_{i=1}^{k}\left(\left(A^n\right)^{i-1}B^n \hat{u}_{k-i}^n\right)\\
&+ \sum_{i=1}^{k}\left(\left(A^n\right)^{i-1}B_w^n w_{k-i}\right) - \left(A^n\right)^k x_0^n\\
&- \sum_{i=1}^{k}\left(\left(A^n\right)^{i-1}B^n u_{k-i}^n\right)\\
&- \sum_{i=1}^{k}\left(\left(A^n\right)^{i-1}B_w^n w_{k-i}\right)\\
=& \sum_{i=1}^{k}\left(\left(A^n\right)^{i-1}B^n \hat{u}_{k-i}^n\right)\\
&- \sum_{i=1}^{k}\left(\left(A^n\right)^{i-1}B^n u_{k-i}^n\right)\\
=& \sum_{i=1}^{k}\left(\left(A^n\right)^{i-1}B^n\left[\hat{u}_{k-i}^n - u_{k-i}^n\right]\right). \quad (32)
\end{aligned}$$

Furthermore, we have

$$\begin{aligned}
C^n\delta_{x,k}^n &= C^n\left(\hat{x}_k^n - x_k^n\right)\\
&= \sum_{i=1}^{k}\left(C^n\left(A^n\right)^{i-1}B^n\left[\hat{u}_{k-i}^n - u_{k-i}^n\right]\right) \quad (33)
\end{aligned}$$

and

$$\begin{aligned}
\left\|C^n\delta_{x,k}^n\right\| &= \left\|\sum_{i=1}^{k}\left(C^n\left(A^n\right)^{i-1}B^n\left[\hat{u}_{k-i}^n - u_{k-i}^n\right]\right)\right\|\\
&\leq \sum_{i=1}^{k}\left\|C^n\left(A^n\right)^{i-1}B^n\left[\hat{u}_{k-i}^n - u_{k-i}^n\right]\right\|\\
&\leq \sum_{i=1}^{k}\left\|C^n\left(A^n\right)^{i-1}B^n\right\|\left\|\hat{u}_{k-i}^n - u_{k-i}^n\right\|\\
&= \sum_{i=1}^{k}\left\|C^n\left(A^n\right)^{i-1}B^n\right\|\left\|\delta_{u,k}^n\right\|\\
&\leq \sum_{i=1}^{k}\delta_u\left\|C^n\left(A^n\right)^{i-1}B^n\right\|\\
&=\delta_u \times \sum_{i=1}^{k}\left\|C^n\left(A^n\right)^{i-1}B^n\right\| = \delta_u M_k^n. \quad (34)
\end{aligned}$$

Now denote the second term of (8) as J_y. Then we have

$$\begin{aligned}
L_y =& J_y(u(\mathcal{W})) - J_y(u(\mathcal{N}))\\
=& \sum_{n=1}^{N}\sum_{k=1}^{p}\left\|C\hat{x}_k^n + b^n\right\|_{Q_y}^2 - \sum_{n=1}^{N}\sum_{k=1}^{p}\left\|C^n x_k^n + b^n\right\|_{Q_y}^2\\
=& \sum_{n=1}^{N}\sum_{k=1}^{p}\left\|C^n(x_k^n + \delta_{x,k}^n) + b^n\right\|_{Q_y}^2\\
&- \sum_{n=1}^{N}\sum_{k=1}^{p}\left\|C^n x_k^n + b^n\right\|_{Q_y}^2\\
=& \sum_{n=1}^{N}\sum_{k=1}^{p}\left(\left\|C^n(x_k^n + \delta_{x,k}^n) + b^n\right\|_{Q_y}^2 - \left\|C^n x_k^n + b^n\right\|_{Q_y}^2\right)\\
=& \sum_{n=1}^{N}\sum_{k=1}^{p}\left(\left\|Q_y^{1/2}C^n(x_k^n + \delta_{x,k}^n) + Q_y^{1/2}b^n\right\|^2 - \left\|Q_y^{1/2}C^n x_k^n + Q_y^{1/2}b^n\right\|^2\right)\\
\leq& \sum_{n=1}^{N}\sum_{k=1}^{p}\Bigg(\left\|Q_y^{1/2}C^n x_k^n + Q_y^{1/2}b^n\right\|^2\\
&+ \left\|Q_y^{1/2}C^n\delta_{x,k}^n\right\|^2 + 2\left\|Q_y^{1/2}C^n x_k^n + Q_y^{1/2}b^n\right\|\\
&\times\left\|Q_y^{1/2}C^n\delta_{x,k}^n\right\| - \left\|Q_y^{1/2}C^n x_k^n + Q_y^{1/2}b^n\right\|^2\Bigg)\\
=& \sum_{n=1}^{N}\sum_{k=1}^{p}\left(\left\|Q_y^{1/2}C^n\delta_{x,k}^n\right\|^2 + 2\left\|Q_y^{1/2}\left(C^n x_k^n + b^n\right)\right\|\right.
\end{aligned}$$

$$
\begin{aligned}
&\times \left\|Q_y^{1/2} C^n \delta_{x,k}^n\right\|\Big) \\
=&\sum_{n=1}^{N}\sum_{k=1}^{p}\left(\left\|Q_y^{1/2} C^n \delta_{x,k}^n\right\|^2\right. \\
&+ 2\left\|Q_y^{1/2}\left(C^n \hat{x}_k^n - C^n \delta_{x,k}^n + b^n\right)\right\| \\
&\left.\times \left\|Q_y^{1/2} C^n \delta_{x,k}^n\right\|\right) \\
\leq&\sum_{n=1}^{N}\sum_{k=1}^{p}\left(\left\|Q_y^{1/2} C^n \delta_{x,k}^n\right\|^2 + 2\left\|Q_y^{1/2} C^n \delta_{x,k}^n\right\|^2\right. \\
&\left.+2\left\|Q_y^{1/2}\left(C^n \hat{x}_k^n + b^n\right)\right\|\left\|Q_y^{1/2} C^n \delta_{x,k}^n\right\|\right) \\
=&\sum_{n=1}^{N}\sum_{k=1}^{p}\left(3\left\|Q_y^{1/2} C^n \delta_{x,k}^n\right\|^2\right. \\
&\left.+2\left\|Q_y^{1/2}\left(C^n \hat{x}_k^n + b^n\right)\right\|\left\|Q_y^{1/2} C^n \delta_{x,k}^n\right\|\right) \\
\leq&\sum_{n=1}^{N}\sum_{k=1}^{p}\Big(3\left\|Q_y\right\|\left\|C^n \delta_{x,k}^n\right\|^2 \\
&+2\left\|Q_y\right\|\left\|C^n \hat{x}_k^n + b^n\right\|\left\|C^n \delta_{x,k}^n\right\|\Big) \\
\leq&\sum_{n=1}^{N}\sum_{k=1}^{p}\Big(3\delta_u^2\left(M_k^n\right)^2\left\|Q_y\right\| \\
&+2\delta_u M_k^n\left\|Q_y\right\|\left\|C^n \hat{x}_k^n + b^n\right\|\Big) \\
=&3\delta_u^2\left\|Q_y\right\|\sum_{n=1}^{N}\sum_{k=1}^{p}\left(M_k^n\right)^2 \\
&+ 2\delta_u\left\|Q_y\right\|\sum_{n=1}^{N}\sum_{k=1}^{p}\left(M_k^n\left\|C^n \hat{x}_k^n + b^n\right\|\right).
\end{aligned} \tag{35}
$$

Putting (31) and (35) together, we have

$$
\begin{aligned}
L(\mathcal{W}) =&L_u + L_y \\
\leq&3pN\delta_u^2\left\|Q_u\right\| + 2\delta_u\left\|Q_u\right\|\sum_{n=1}^{N}\sum_{k=0}^{p-1}\left(\left\|\hat{u}_k^n\right\|\right) \\
&+ 3\delta_u^2\left\|Q_y\right\|\sum_{n=1}^{N}\sum_{k=1}^{p}\left(M_k^n\right)^2 \\
&+ 2\delta_u\left\|Q_y\right\|\sum_{n=1}^{N}\sum_{k=1}^{p}\left(M_k^n\left\|C^n \hat{x}_k^n + b^n\right\|\right).
\end{aligned}
$$

This completes the proof. ■

References

[1] D. Görges, "Distributed adaptive linear quadratic control using distributed reinforcement learning," *IFAC-PapersOnLine*, vol. 52, no. 11, pp. 218–223, 2019.

[2] 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.

[3] Y. Zheng, S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick, "Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 899–910, 2016.

[4] P. D. Christofides, R. Scattolini, D. M. de la Pena, and J. Liu, "Distributed model predictive control: A tutorial review and future research directions," *Computers & Chemical Engineering*, vol. 51, pp. 21–41, 2013.

[5] A. Ma, K. Liu, Q. Zhang, and Y. Xia, "Distributed MPC for linear discrete-time systems with disturbances and coupled states," *Systems & Control Letters*, vol. 135, pp. 104 578–104 585, 2020.

[6] R. Negenborn and J. Maestre, "On 35 approaches for distributed MPC made easy," in *Distributed model predictive control made easy*. Springer, 2014, pp. 1–37.

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

[8] G. Darivianakis, A. Eichler, and J. Lygeros, "Distributed model predictive control for linear systems with adaptive terminal sets," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 1044–1056, 2019.

[9] D. Jia and B. H. Krogh, "Distributed model predictive control," in *Proceedings of the 2001 American Control Conference*, Arlington, VA, June 25-27, 2001, pp. 2767–2772.

[10] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.

[11] P. Graf, J. Annoni, C. Bay, D. Biagioni, D. Sigler, M. Lunacek, and W. Jones, "Distributed reinforcement learning with ADMM-RL," in *2019 American Control Conference*, Philadelphia, PA, July 10-19, 2019, pp. 4159–4166.

[12] P. Varshavskaya, L. P. Kaelbling, and D. Rus, "Efficient distributed reinforcement learning through agreement," in *Distributed Autonomous Robotic Systems 8*. Springer, 2009, pp. 367–378.

[13] K. Kosaraju, S. Sivaranjani, W. Suttle, V. Gupta, and J. Liu, "Reinforcement learning based distributed control of dissipative networked systems," *arXiv preprint arXiv:2011.14263*, 2020.

[14] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1086–1095, 2019.

[15] M. Kale and A. Chipperfield, "Stabilized mpc formulations for robust reconfigurable flight control," *Control Engineering Practice*, vol. 13, no. 6, pp. 771–788, 2005.

[16] T. Bai, S. Li, and Y. Zou, "Distributed mpc for reconfigurable architecture systems via alternating direction method of multipliers," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 7, pp. 1336–1344, 2020.

[17] D. J. Burns, C. Danielson, J. Zhou, and S. Di Cairano, "Reconfigurable model predictive control for multievaporator vapor compression systems," *IEEE Trans. Control Syst. Techn.*, vol. 26, no. 3, pp. 984–1000, 2017.

[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] 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.

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

[21] J. Xu, B. Cao, S. Li, B. Wang, and B. Ning, "A hybrid criterion based balancing strategy for battery energy storage systems," *Energy Procedia*, vol. 103, pp. 225–230, 2016.

[22] 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*, vol. 2017, no. Article ID 6510747, 2017.

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