Reconfigurable Model Predictive Control for Large Scale Distributed Systems

Jun Chen, *Senior Member, IEEE*, Lei Zhang and Weinan Gao, *Senior Member, IEEE*

***Abstract*—For large scale distributed systems, centralized model predictive control (MPC) often requires high computational resources, while distributed MPC can only achieve suboptimal control performance if the computation resource is limited. To address these limitations, this paper proposes a new reconfigurable MPC framework for large scale distributed systems, in which an optimal control problem with a time-varying structure is formulated and solved for each control loop. More specifically, at each time step, a subset of the control inputs is dynamically selected to be optimized by MPC, while the previous optimal solution is applied to the remaining control inputs. A theoretical upper bound on the performance loss, due to the fact that only a subset of inputs is optimized, is then derived to guarantee the worst-case performance. To minimize the performance loss, this upper bound is then used to guide the reconfiguration of MPC, i.e., the selection of control inputs for optimization. The applicability of the proposed approach is illustrated through case studies, including battery cell-to-cell balancing control and multi-vehicle formation control. Numerical results confirm that the proposed approach can achieve better control performance than distributed MPC and requires less computation time than conventional centralized MPC.**

***Index Terms*—Model predictive control, distributed systems, suboptimality, reconfigurable control, battery, formation control.**

I. Introduction

Control of large distributed systems is of prominent importance for many applications [1]–[6]. Among many approaches, model predictive control (MPC) has been extensively investigated [7]–[15]. For large scale systems, distributed MPC has been widely used in [16]–[21], which can be grouped into non-cooperative distributed MPC, cooperative distributed MPC, and decomposed optimization approach [16]. For example, the work [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. The work [22] 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. The work [21] studies the distributed MPC without a centralized 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. Finally, [23] studies the conditions under which distributed MPC can achieve centralized-like performance, which requires a large number of iterations before converging to the global optimum.

Despite the promising results discussed above, distributed MPC can only achieve suboptimal control performance (when the computation resource is limited) while requiring high communication resources [16]. On the other hand, centralized MPC has the advantage of achieving optimality and reducing communication among agents, and therefore has been widely researched [14], [15], [24]–[27]. However, centralized MPC usually requires more significant computational resources than distributed MPC, and hence intractable for large scale systems. To address these issues, this paper proposes a new reconfigurable MPC (ReMPC) 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 control inputs is dynamically selected to be optimized by MPC, while the previous optimal solution is applied to the remaining control inputs. Note that since the OCP is reconfigured in real-time, the set of control inputs to be optimized is time-varying and is chosen based on real-time feedback and a predefined reconfiguration policy. Such approach effectively reduces the computational requirement of MPC, as the number of optimization variables are significantly reduced.

On the other hand, the proposed ReMPC framework can only achieve suboptimal control performance since control authority is reduced. To quantify the performance loss, a theoretical upper bound is derived to guarantee the worst case control performance. Furthermore, this upper bound is in turn used to guide the reconfiguration of MPC so that the performance loss is minimized. The applicability of the proposed approach is illustrated through practical examples, including (i) battery cell-to-cell balancing control problem,

This work is supported in part by SECS Faculty Startup Fund at Oakland University, in part by Michigan Space Grant Consortium grant #80NSSC20M0124, and in part by National Science Foundation through Award #2237317.

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

Lei Zhang is with the National Engineering Research Center for Electric Vehicles, Beijing Institute of Technology, Beijing 100081, China (email: lei_zhang@bit.edu.cn).

Weinan Gao is with the Department of Mechanical and Civil Engineering, Florida Institute of Technology, Melbourne, FL 32901, USA (email: weinan.gao@nyu.edu)

Jun Chen is the corresponding author.

where the system has 100 inputs to be optimized and (ii) multi-vehicle formation control problem. Numerical results confirm that the proposed ReMPC can achieve better control performance compared to distributed MPC and requires less computation compared to conventional centralized MPC.

A similar concept of optimizing over a subset of the control inputs to reduce computational requirements has been introduced in the literature. For example, the work [28] proposes channel-hopping MPC where only one control input is optimized for each time step. However, such an approach poses two issues. Firstly, since only one input is optimized, control performance can be largely degraded due to a significant loss of control authority. Secondly, the channel-hopping MPC proposed in [28] requires solving multiple optimization problems, one for each control input, and implement only the best one. Therefore, the number of optimization problems being solved at each time step is the same as that of control inputs, resulting high computational requirement if the number of control inputs is high. The proposed ReMPC framework is different from and more general than channel-hopping MPC in [28]. In ReMPC, at each time step, only one optimization problem will be solved, resulting in less computation. In addition, multiple control inputs can be simultaneously optimized at each time step, leading to less optimality loss. The proposed ReMPC is also different from event-triggered MPC [24], [29]–[35], where an OCP optimizing all control inputs is formulated and solved only when an event is triggered. Firstly, control inputs are optimized aperiodically but synchronously in event-triggered MPC, while the optimization of control inputs is both aperiodic and asynchronous in ReMPC. Secondly, event-triggered MPC can reduce the average computation time by reducing the number of optimization instances, but the worst-case computation remains the same. On the other hand, the proposed ReMPC can substantially reduce both average and worst-case computation time, since a smaller OCP is solved for each time step.

The proposed reconfigurable MPC (ReMPC) framework is also different from those in [36]–[38], where the notion of “reconfigurable MPC” is used for an MPC control strategy where the physical plant is reconfigurable. For example, [37] considers MPC for linear systems with changeable network topology, and proposes a novel reconfiguration control scheme based on ADMM (alternating direction method of multipliers). The work [38] applies MPC to multievaporator vapor compression systems, where individual evaporators can be turned on or off. An MPC is then designed to accommodate the time varying system configuration. In other words, the reconfigurable MPC considered in [36]–[38] mainly refers to the fact that the system under control can change structures in real-time, and therefore MPC is reconfigured accordingly. On the other hand, in the proposed ReMPC framework, the physical systems are assumed to be fixed, but MPC dynamically selects a subset of the control inputs to form OCP to reduce the required online computations. The novel contribution of this paper can be summarized as follows.

1) A reconfigurable MPC (ReMPC) framework is proposed, in which MPC optimizes over a subset of control inputs to reduce computation. As this subset is dynamically selected in real-time, all control inputs are still being updated based on measurement feedback, but asynchronously with heterogeneous sampling time.
2) An upperbound on the optimality loss compared to optimization over all contol inputs is derived to guarantee an acceptable worst-case performance.
3) A reconfiguration policy is developed such that the optimality loss is minimized.
4) The effectiveness of the proposed ReMPC framework is demonstrated through practical examples including battery cell balancing control and multi-vehicle formation control.

The rest of this paper is organized as follows. Section II presents the proposed reconfigurable MPC, while theoretical guarantees on performance loss and loss-based reconfiguration strategy are discussed in Section III. Numerical simulation results on cell-to-cell balancing control of 100 connected cells and multi-vehicle formation control are presented in Section IV. The paper is concluded in Section V.

Notations: Throughout the paper, we make use of the following notations and properties. We use $\|\cdot\|$ without subscript to denote 2-norm of a vector or matrix. 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*

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

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. Reconfigurable Model Predictive Control

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

$$x_{k+1}^n = A^n x_k^n + B^n u_k^n \tag{1a}$$

$$y_k^n = C^n x_k^n + b^n, \qquad n \in \mathcal{N} \tag{1b}$$

where $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of all distributed components, x^n, u^n and y^n are the states, outputs and inputs for nth subsystem. Denote n_x, n_y and n_u as the number of states, outputs and inputs for each distributed component, respectively. A^n, B^n and C^n are system matrices and b^n is the affine term, all with proper dimension. Furthermore, the inputs and outputs of each components are coupled through constraints, as follows:

$$\{u_k^1, u_k^2, \ldots, u_k^n\} \in \mathcal{U} \subseteq R^{N n_u} \tag{2a}$$

$$\{y_k^1, y_k^2, \ldots, y_k^n\} \in \mathcal{Y} \subseteq R^{N n_y}, \tag{2b}$$

Remark 1. *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, given current state estimate $\tilde{x}^n$, $n \in \mathcal{N}$, MPC solves the following optimal control problem (OCP) over a prediction horizon p:

$$\min_{u_k^n, n\in\mathcal{N}} \quad J = \sum_{n=1}^{N}\sum_{k=1}^{p} ||y_k^n||_{Q_y}^2 + \sum_{n=1}^{N}\sum_{k=0}^{p-1} ||u_k^n||_{Q_u}^2 \tag{3a}$$

$$\text{s.t.} \quad \text{system dynamics (1)}, \qquad \forall n \in \mathcal{N} \tag{3b}$$

$$x_0^n = \tilde{x}^n, \qquad \forall n \in \mathcal{N} \tag{3c}$$

$$\text{input and output constraints (2)}, \ \forall k = 0, 1, \ldots, p. \tag{3d}$$

Note that the weight matrixs Q_y is assumed to be symmetric and positive semidefinite and Q_u is assumed to be symmetric and positive definite. It is then trivial to see that the total number of optimization variables is Npn_u. When N is large, solving the above OCP (3) is intractable (even for small prediction horizon p) due to the high computational requirement. To address this issue, in this paper, a reconfigurable MPC framework is proposed where a subset of components is dynamically selected to form the OCP, while for the remaining components, previous optimal solution is applied as control inputs. In the sequel, we will discuss in detail the formal formulation of such OCP with only a subset of components.

Given a subset $\mathcal{W} \subseteq \mathcal{N}$, denote its complementary set as $\overline{\mathcal{W}} = \mathcal{N} - \mathcal{W}$. To formally present 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 an input sequence $\bar{u}^n$, define $x^n(\bar{u}^n)$ 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. *At any time step k, an input sequence $\bar{u}^n = \begin{bmatrix}\bar{u}_0^n & \bar{u}_1^n & \cdots & \bar{u}_{p-1}^n\end{bmatrix}^T$ is available for all $n \in \overline{\mathcal{W}}$.*

Remark 2. *Assumption 1 implies that there exists a control input for components in $\overline{\mathcal{W}}$, which is not necessarily optimal. This is not a restrictive assumption, since one can always set $\mathcal{W} = \mathcal{N}$ and solve the full OCP (3) at initialization. This will make control inputs available for all components for p time steps. More specifically, let $\bar{u}^{n,*} \begin{bmatrix}\bar{u}_0^{n,*} & \bar{u}_1^{n,*} & \cdots & \bar{u}_{p-1}^{n,*}\end{bmatrix}^T$ be the input sequence from the last time step, then we can set $\bar{u}^n$ for the current time step as*

$$\bar{u}^n = \begin{bmatrix}\bar{u}_1^{n,*} & \bar{u}_2^{n,*} & \cdots & \bar{u}_{p-1}^{n,*} & \bar{u}_{p-1}^{n,*}\end{bmatrix}^T. \tag{4}$$

In other words, for $n \in \overline{\mathcal{W}}$, input sequence $\bar{u}^n$ can be obtained by shifting the previous input sequence $\bar{u}^{n,*}$ by one and apply the zero order to the last element. Note that if nth component was optimized in the previous time step, then $\bar{u}^{n,*}$ is in fact given by solving the reduced size MPC, as detailed below.

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 $\hat{\mathcal{U}}$ and the set of feasible output for $n \in \mathcal{W}$ as $\hat{\mathcal{Y}}$.*

Now we are ready to formulate a reduced size OCP that only includes components in $\mathcal{W}$. At each time step, given current state estimate $\tilde{x}^n$, $n = 1, \ldots, N$, the following reduced size OCP is formulated:

$$\min_{u_k^n, n\in\mathcal{W}} \quad J = \sum_{n\in\mathcal{W}}\sum_{k=1}^{p} ||y_k^n||_{Q_y}^2 + \sum_{n\in\mathcal{W}}\sum_{k=0}^{p-1} ||u_k^n||_{Q_u}^2 \tag{5a}$$

$$\text{s.t.} \quad \text{system dynamics (1)}, \qquad \forall n \in \mathcal{W} \tag{5b}$$

$$x_0^n = \tilde{x}^n, \qquad \forall n \in \mathcal{W} \tag{5c}$$

$$\{u_k^n \mid n \in \mathcal{W}\} \in \hat{\mathcal{U}}, \qquad \forall k \tag{5d}$$

$$\{y_k^n \mid n \in \mathcal{W}\} \in \hat{\mathcal{Y}}, \qquad \forall k. \tag{5e}$$

Algorithm 1: Reconfigurable Model Predictive Control

1 Initialize by solving MPC($\mathcal{N}$), i.e., standard OCP (3) for all components, to get optimal solution $\hat{u}^n$ for all $n \in \mathcal{N}$;
2 **for** $n \in \mathcal{N}$ **do**
3 $\quad$ $\bar{u}^n \leftarrow \hat{u}^n$; % Store solution
4 **end**
5 Apply first control move $\hat{u}_0^n$ and move to next time step;
6 **while** $t \le T$ **do**
7 $\quad$ Collect current state estimate $\tilde{x}^n$;
8 $\quad$ Select a new $\mathcal{W} \subseteq \mathcal{N}$;
9 $\quad$ $\hat{u}^n \leftarrow$ Solve MPC($\mathcal{W}$) as formulated by (5);
10 $\quad$ **for** $n \in \mathcal{W}$ **do**
11 $\quad\quad$ $\bar{u}^n \leftarrow \hat{u}^n$; % Store solution
12 $\quad$ **end**
13 $\quad$ **for** $n \in \overline{\mathcal{W}}$ **do**
14 $\quad\quad$ $\hat{u}_0^n \leftarrow$ (4); % Shifting previous optimal solution
15 $\quad$ **end**
16 $\quad$ Apply $\hat{u}_0^n$ for all $n \in \mathcal{N}$ and move to next time step;
17 **end**

For each control loop, the proposed reconfigurable MPC selects $\mathcal{W}$, solves OCP (5), and assembles the control vector $u(\mathcal{W}) = \{\hat{u}_k^n\}$ according to the following.

$$\hat{u}^n = \begin{cases} \text{solution of (5)} & \text{if } n \in \mathcal{W} \\ \bar{u}^n \text{ as defined in (4)} & \text{if } n \in \overline{\mathcal{W}}. \end{cases} \tag{6}$$

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($\mathcal{N}$). Algorithm 1 formally presents the proposed ReMPC framework, where $\hat{u}_0^n$ denotes the first element in $\hat{u}^n$. As can be seen, all control inputs are optimized at initialization at Line 1 and their solution stored at Line 2–4, fulfilling Assumption 1 for all subsequent steps. Then for each time step, a new subset $\mathcal{W}$ is selected at Line 8 to form the reduced OCP (5), which is also termed as MPC($\mathcal{W}$) and solved at Line 9. The latest optimal control sequence for $n \in \mathcal{W}$ is then stored in memory at Line 10–12, while for $n \in \overline{\mathcal{W}}$, i.e., components not selected for optimization, their previous optimal solution (as saved in memory) is shifted to obtain $\hat{u}^n$ at Line 13–15. Line 16 applies the first control move for each component and move to the next time step.

Remark 3. *Note that MPC($\mathcal{W}$) only optimizes control inputs for components in $\mathcal{W}$, while for $n \in \overline{\mathcal{W}}$, previous optimal solution is used to implement its control, which is also used to form the constraints in (5d) and (5e). Therefore, the number of optimization variables of MPC($\mathcal{W}$) is reduced to $|\mathcal{W}|pn_u$.*

Remark 4. *As can be seen from Line 8 of Algorithm 1, a new subset $\mathcal{W}$ is selected at each time step. The notion of $\mathcal{W}$ instead of $\mathcal{W}_k$ is used for the simplicity of notation, i.e., we drop the subscript k. It should also be noted that since $\mathcal{W}$ is varying, its selection can guarantee that all control inputs are updated using measurement feedback, but with heterogeneous and aperiodic sampling time.*

To ensure that OCP (5) 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 $\hat{\mathcal{Y}} \neq \emptyset$.*

Remark 5. *Assumption 2 guarantees that the feasibility of MPC($\mathcal{W}$) for each time step, regardless of the choice of $\mathcal{W}$. This could be a restrictive assumption if time-varying constraints are considered or $\bar{u}^n$ is arbitrarily selected. However, in this paper, we only consider time invariant constraints, i.e., $\mathcal{U}$ and $\mathcal{Y}$ of (2) are time-invariant. By using previous optimized control sequence for $n \in \overline{\mathcal{W}}$, as detailed in Remark 2, the assumption that $\hat{\mathcal{U}} \neq \emptyset$ always holds. However, $\hat{\mathcal{Y}} \neq \emptyset$ may not always hold. In this case, one can use soft output constraint as often done in practice [39].*

Remark 6. *Comparing the proposed ReMPC as presented in Algorithm 1 to the channel-hopping MPC discussed in [28], the proposed ReMPC has several advantages. Firstly, in channel-hopping MPC, only one control input is optimized at each time step, while ReMPC can optimize multiple control inputs when $|\mathcal{W}| > 1$. Secondly, channel-hopping MPC requires solving multiple optimization problems, one for each control input, while ReMPC only performs optimization solving once for each time step, as can be seen from Line 9 of Algorithm 1.*

Remark 7. *The proposed ReMPC possesses several similarities to event-triggered MPC [24], [30]–[33], [35], [40], where an OCP optimizing all control inputs is formulated and solved only when an event is triggered. Firstly, control inputs are optimized aperiodically in both event-triggered MPC and ReMPC. Secondly, both event-triggered MPC and ReMPC can reduce computational requirement significantly. However, the proposed ReMPC as presented in Algorithm 1 is substantially different from event-triggered. Though control inputs are optimized aperiodically in event-triggered MPC, they will be optimized all together whenever an event is triggered. However, in ReMPC, control inputs are optimized aperiodically and asynchronously. Furthermore, though event-triggered MPC can save average computation time, the worst-case computation time remains unchanged, as an OCP with all control inputs needs to be solved whenever an event is triggered. However, for the proposed ReMPC, both average and worst-case computation time are substantially decreased, since a smaller OCP is solved for each time step.*

III. Performance Loss and Subsection Selection

Algorithm 1 presents the proposed ReMPC framework in its generic form. Now we need to address the following two 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?

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 $u = \{u^n\}$, $n \in \mathcal{N}$, with a slight abuse of notation, define the following performance index,

$$J(u) = \sum_{n=1}^{N}\sum_{k=1}^{p} ||y_k^n(u_k^n)||_{Q_y}^2 + \sum_{n=1}^{N}\sum_{k=0}^{p-1} ||u_k^n||_{Q_u}^2. \tag{7}$$

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

$$L(\mathcal{W}) = J(u(\mathcal{W})) - J(u(\mathcal{N})). \tag{8}$$

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

Lemma 1. *Given system (1) and performance index (7), $J(u(\mathcal{N}))$ is upperbounded by*

$$J(u(\mathcal{N})) \leq Np\,\|Q_y\|\,\|\Delta_y\|^2 + Np\,\|Q_u\|\,\|\Delta_u\|^2, \tag{9}$$

where

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

Proof.

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

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.,*

$$\delta_u = \max_{k,n} \left\|\delta_{u,k}^n\right\| = \max_{k,n} \|u_k^n - \hat{u}_k^n\|. \tag{10}$$

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

$$M_k^n = \sum_{i=1}^{k} \left\| C^n \left(A^n\right)^{i-1} B^n \right\|. \tag{11}$$

Then the following theorem provides an upperbound for $L(\mathcal{W}) = J(u(\mathcal{W})) - J(u(\mathcal{N}))$, i.e., an analytical quantification of the performance loss for a given $\mathcal{W}$, which will be used in the next section, as a criterion to select $\mathcal{W}$ such that the performance loss is minimized.

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

$$\begin{aligned} L(\mathcal{W}) \leq & 3pN\delta_u^2 \left\|Q_u\right\| + 2\delta_u \left\|Q_u\right\| \sum_{n=1}^{N}\sum_{k=0}^{p-1} \left(\left\|\hat{u}_k^n\right\|\right) \\ & + 3\delta_u^2 \left\|Q_y\right\| \sum_{n=1}^{N}\sum_{k=1}^{p} \left(M_k^n\right)^2 \\ & + 2\delta_u \left\|Q_y\right\| \sum_{n=1}^{N}\sum_{k=1}^{p} \left(M_k^n \left\|C^n \hat{x}_k^n + b^n\right\|\right). \end{aligned} \tag{12}$$

Proof. Denote the second term of (7) as J_u. Then we have

$$\begin{aligned} L_u =& J_u(u(\mathcal{W})) - J_u(u(\mathcal{N})) \\ =& \sum_{n=1}^{N}\sum_{k=0}^{p-1} \left(\left\| Q_u^{1/2} u_k^n + Q_u^{1/2}\delta_{u,k}^n \right\|^2 - \left\| Q_u^{1/2} u_k^n \right\|^2 \right) \\ \leq& \sum_{n=1}^{N}\sum_{k=0}^{p-1} \left(\left\| Q_u^{1/2} u_k^n \right\|^2 + \left\| Q_u^{1/2}\delta_{u,k}^n \right\|^2 \right. \\ & \left. + 2\left\| Q_u^{1/2} u_k^n \right\| \left\| Q_u^{1/2}\delta_{u,k}^n \right\| - \left\| Q_u^{1/2} u_k^n \right\|^2 \right) \\ =& \sum_{n=1}^{N}\sum_{k=0}^{p-1} \left(\left\| Q_u^{1/2}\delta_{u,k}^n \right\|^2 \right. \\ & \left. + 2\left\| Q_u^{1/2} u_k^n \right\| \left\| Q_u^{1/2}\delta_{u,k}^n \right\| \right) \\ =& \sum_{n=1}^{N}\sum_{k=0}^{p-1} \left(\left\| Q_u^{1/2}\delta_{u,k}^n \right\|^2 \right. \\ & \left. + 2\left\| Q_u^{1/2} \hat{u}_k^n - Q_u^{1/2}\delta_{u,k}^n \right\| \left\| Q_u^{1/2}\delta_{u,k}^n \right\| \right) \\ \leq& \sum_{n=1}^{N}\sum_{k=0}^{p-1} \left(\left\| Q_u^{1/2}\delta_{u,k}^n \right\|^2 + 2\left[\left\| Q_u^{1/2} \hat{u}_k^n \right\| \right. \right. \\ & \left. \left. + \left\| Q_u^{1/2}\delta_{u,k}^n \right\| \right] \times \left\| Q_u^{1/2}\delta_{u,k}^n \right\| \right) \\ =& \sum_{n=1}^{N}\sum_{k=0}^{p-1} \left(3\left\| Q_u^{1/2}\delta_{u,k}^n \right\|^2 \right. \\ & \left. + 2\left\| Q_u^{1/2} \hat{u}_k^n \right\| \left\| Q_u^{1/2}\delta_{u,k}^n \right\| \right) \\ \leq& \sum_{n=1}^{N}\sum_{k=0}^{p-1} \left(3\left\|Q_u\right\| \left\|\delta_{u,k}^n\right\|^2 + 2\left\|Q_u\right\| \left\|\hat{u}_k^n\right\| \left\|\delta_{u,k}^n\right\| \right) \\ \leq& \sum_{n=1}^{N}\sum_{k=0}^{p-1} \left(3\delta_u^2 \left\|Q_u\right\| + 2\delta_u \left\|Q_u\right\| \left\|\hat{u}_k^n\right\| \right) \\ =& 3pN\delta_u^2 \left\|Q_u\right\| + 2\delta_u \left\|Q_u\right\| \sum_{n=1}^{N}\sum_{k=0}^{p-1} \left(\left\|\hat{u}_k^n\right\|\right). \end{aligned} \tag{13}$$

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

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

Furthermore, we have

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

and

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

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

Now denote the first term of (7) 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} \| C \hat{x}_k^n + b^n \|_{Q_y}^2 - \sum_{n=1}^{N} \sum_{k=1}^{p} \| C^n x_k^n + b^n \|_{Q_y}^2 \\
=& \sum_{n=1}^{N} \sum_{k=1}^{p} \| C^n (x_k^n + \delta_{x,k}^n) + b^n \|_{Q_y}^2 \\
&- \sum_{n=1}^{N} \sum_{k=1}^{p} \| C^n x_k^n + b^n \|_{Q_y}^2 \\
=& \sum_{n=1}^{N} \sum_{k=1}^{p} \Big(\| C^n (x_k^n + \delta_{x,k}^n) + b^n \|_{Q_y}^2 \\
&\qquad - \| C^n x_k^n + b^n \|_{Q_y}^2 \Big) \\
=& \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 \right. \\
&\qquad \left. - \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} \left(\left\| Q_y^{1/2} C^n x_k^n + Q_y^{1/2} b^n \right\|^2 \right. \\
&+ \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\| \\
&\left. \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 \right) \\
=& \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. \\
&\qquad \left. \times \left\| Q_y^{1/2} C^n \delta_{x,k}^n \right\| \right) \\
=& \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 \| Q_y \| \| C^n \delta_{x,k}^n \|^2 \\
&+ 2 \| Q_y \| \| C^n \hat{x}_k^n + b^n \| \| C^n \delta_{x,k}^n \| \Big) \\
\leq& \sum_{n=1}^{N} \sum_{k=1}^{p} \Big(3 \delta_u^2 \left(M_k^n \right)^2 \| Q_y \| \\
&+ 2 \delta_u M_k^n \| Q_y \| \| C^n \hat{x}_k^n + b^n \| \Big) \\
=& 3 \delta_u^2 \| Q_y \| \sum_{n=1}^{N} \sum_{k=1}^{p} \left(M_k^n \right)^2 \\
&+ 2 \delta_u \| Q_y \| \sum_{n=1}^{N} \sum_{k=1}^{p} \left(M_k^n \| C^n \hat{x}_k^n + b^n \| \right).
\end{aligned} \tag{14}$$

Putting (13) and (14) together, we have

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

This completes the proof. □

Remark 8. *Note that comparing the upperbound* (12) *of the performance loss* $L(\mathcal{W})$ *with the upperbound (9) of the performance index* $J(u(\mathcal{N}))$*, it is apparent that (12) is useful only when* δ_u *is sufficiently small. Otherwise* (12) *can become overly conservative. In the sequel, we assume that a meaningful* δ_u *can be determined through the input constraints. Note that this is not a restrictive assumption, since in many practical applications, rate constraints are often applied due to the physical limits of actuators.*

Theorem 1 provides an upperbound for the performance loss for a given $\mathcal{W}$. In the next section, we will make use of this upperbound to select $\mathcal{W}$ such that the performance loss is minimized.

B. Loss-based Reconfiguration

To answer (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 two positive integers* $d_1 < d_2 \leq N$*, let*

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

Then we have

$$L(\mathcal{W}_1) \geq L(\mathcal{W}_2). \tag{16}$$

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 $|\mathcal{W}_3| = d_2$, where $|\cdot|$ denotes the set cardinality.

Denote the solution of MPC($\mathcal{W}_1$) as $\{\hat{u}^n_{1,k}\}$, and the solution of MPC($\mathcal{W}_3$) as $\{\hat{u}^n_{3,k}\}$. Then we have

$$
\begin{aligned}
J(u(\mathcal{W}_3)) =& \sum_{n=1}^{N}\sum_{k=1}^{p}||y^n_k(\hat{u}^n_{3,k})||^2_{Q_y} + \sum_{n=1}^{N}\sum_{k=0}^{p-1}||\hat{u}^n_{3,k}||^2_{Q_u} \\
=& \sum_{n\in\mathcal{W}_1}\sum_{k=1}^{p}||y^n_k(\hat{u}^n_{3,k})||^2_{Q_y} + \sum_{n\in\mathcal{W}_1}\sum_{k=0}^{p-1}||\hat{u}^n_{3,k}||^2_{Q_u} \\
&+ \sum_{n\in\mathcal{W}_3-\mathcal{W}_1}\sum_{k=1}^{p}||y^n_k(\hat{u}^n_{3,k})||^2_{Q_y} \\
&+ \sum_{n\in\mathcal{W}_3-\mathcal{W}_1}\sum_{k=0}^{p-1}||\hat{u}^n_{3,k}||^2_{Q_u} \\
&+ \sum_{n\notin\mathcal{W}_3}\sum_{k=1}^{p}||y^n_k(\hat{u}^n_{3,k})||^2_{Q_y} \\
&+ \sum_{n\notin\mathcal{W}_3}\sum_{k=0}^{p-1}||\hat{u}^n_{3,k}||^2_{Q_u} \\
=& \sum_{n\in\mathcal{W}_1}\sum_{k=1}^{p}||y^n_k(\hat{u}^n_{3,k})||^2_{Q_y} + \sum_{n\in\mathcal{W}_1}\sum_{k=0}^{p-1}||\hat{u}^n_{3,k}||^2_{Q_u} \\
&+ \sum_{n\in\mathcal{W}_3-\mathcal{W}_1}\sum_{k=1}^{p}||y^n_k(\hat{u}^n_{3,k})||^2_{Q_y} \\
&+ \sum_{n\in\mathcal{W}_3-\mathcal{W}_1}\sum_{k=0}^{p-1}||\hat{u}^n_{3,k}||^2_{Q_u} \\
&+ \sum_{n\notin\mathcal{W}_3}\sum_{k=1}^{p}||y^n_k(u^n)||^2_{Q_y} + \sum_{n\notin\mathcal{W}_3}\sum_{k=0}^{p-1}||u^n||^2_{Q_u} \\
\leq& \sum_{n\in\mathcal{W}_1}\sum_{k=1}^{p}||y^n_k(\hat{u}^n_{1,k})||^2_{Q_y} + \sum_{n\in\mathcal{W}_1}\sum_{k=0}^{p-1}||\hat{u}^n_{1,k}||^2_{Q_u} \\
&+ \sum_{n\in\mathcal{W}_3-\mathcal{W}_1}\sum_{k=1}^{p}||y^n_k(u^n)||^2_{Q_y} \\
&+ \sum_{n\in\mathcal{W}_3-\mathcal{W}_1}\sum_{k=0}^{p-1}||u^n||^2_{Q_u} \\
&+ \sum_{n\notin\mathcal{W}_3}\sum_{k=1}^{p}||y^n_k(u^n)||^2_{Q_y} + \sum_{n\notin\mathcal{W}_3}\sum_{k=0}^{p-1}||u^n||^2_{Q_u} \\
=&J(u(\mathcal{W}_1)).
\end{aligned}
$$

Now we have

$$
\begin{aligned}
L(\mathcal{W}_1) =&J(u(\mathcal{W}_1)) - J(u(\mathcal{N})) \\
\geq&J(u(\mathcal{W}_3)) - J(u(\mathcal{N}))) \\
=&L(\mathcal{W}_3).
\end{aligned} \tag{17}
$$

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. □

The following Lemma is a direct result of Lemma 2.

Lemma 3. *Given $\mathcal{W}_1$ and $\mathcal{W}_2$ such that $\mathcal{W}_1 \subset \mathcal{W}_2$, then $L(\mathcal{W}_1) \geq L(\mathcal{W}_2)$.*

According to Lemma 2 or Lemma 3, it is obvious that

$$\arg\min_{\mathcal{W}} L(\mathcal{W}) = \arg\min_{\mathcal{W},|\mathcal{W}|=N} L(\mathcal{W}) = \mathcal{N}, \tag{18}$$

i.e., $\mathcal{N}$ is the solution to (15), making (15) a trivial criteria for selecting a set $\mathcal{W}$. 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,

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

where d is predefined to balance control performance and computation. Note that at each time step, we want to select $\mathcal{W}$ prior to solving any OCP. However, utilizing (19) 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. We first define $\delta_{u,\emptyset}$ according to Definition 3 with $\mathcal{W} = \emptyset$. In other words, $\delta_{u,\emptyset}$ is the maximum change on control inputs if all inputs are selected for optimization, and therefore $\delta_u \leq \delta_{u,\emptyset}$ for all $\mathcal{W}$. Note that calculating $\delta_{u,\emptyset}$ requires solving MPC($\mathcal{N}$). However, in practice, $\delta_{u,\emptyset}$ can also be relaxed to be the rate constraints on control inputs, which can be a constant value in many applications. Define

$$
\begin{aligned}
L_{\mathcal{W}} =&2\delta_u \left\|Q_u\right\| \sum_{n\in\mathcal{W}}\sum_{k=0}^{p-1}(||\hat{u}^n_k||) \\
&+ 3\delta_u^2 \left\|Q_y\right\| \sum_{n\in\mathcal{W}}\sum_{k=1}^{p}(M^n_k)^2 \\
&+ 2\delta_u \left\|Q_y\right\| \sum_{n\in\mathcal{W}}\sum_{k=1}^{p}(M^n_k \left\|C^n\hat{x}^n_k + b^n\right\|) \\
L_{\overline{\mathcal{W}}} =&2\delta_{u,\emptyset} \left\|Q_u\right\| \sum_{n\in\overline{\mathcal{W}}}\sum_{k=0}^{p-1}(||\bar{u}^n_k||) \\
&+ 3\delta_{u,\emptyset}^2 \left\|Q_y\right\| \sum_{n\in\overline{\mathcal{W}}}\sum_{k=1}^{p}(M^n_k)^2 \\
&+ 2\delta_{u,\emptyset} \left\|Q_y\right\| \sum_{n\in\overline{\mathcal{W}}}\sum_{k=1}^{p}(M^n_k \left\|C^n\bar{x}^n_k + b^n\right\|).
\end{aligned}
$$

Then the upper bound given in (12) on performance loss $L(\mathcal{W})$ can be relaxed to

$$L(\mathcal{W}) \leq 3pN\delta_{u,\emptyset}^2 \left\|Q_u\right\| + L_{\mathcal{W}} + L_{\overline{\mathcal{W}}}. \tag{20}$$

Here the first term on the right side of (20) 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 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

$$\mathcal{W} = \arg \min_{\mathcal{W}, |\mathcal{W}|=d} L_{\overline{\mathcal{W}}}. \tag{21}$$

To solve (21), we first rewrite $L_{\overline{\mathcal{W}}}$ as

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

where

$$\begin{aligned} L_n =& 2\delta_{u,\emptyset} \|Q_u\| \sum_{k=0}^{p-1} (\|\bar{u}_k^n\|) + 3\delta_{u,\emptyset}^2 \|Q_y\| \sum_{k=1}^{p} (M_k^n)^2 \\ &+ 2\delta_{u,\emptyset} \|Q_y\| \sum_{k=1}^{p} (M_k^n \|C^n \bar{x}_k^n + b^n\|) . \end{aligned} \tag{22}$$

Given $\{L_n\}$ where $n \in \mathcal{N}$, define the set of d largest elements of $\{L_n\}$ as $\texttt{max\_d}(\{L_n\})$. Utilizing the fact that L_n for each n is independent of each other, we have

$$\mathcal{W} = \{n \in \mathcal{W} | L_n \in \texttt{max\_d}(\{L_n\})\} . \tag{23}$$

Remark 9. *The equation (19) selects $\mathcal{W}$ that minimizes the control performance loss. However, this would require solving MPC($\mathcal{W}$) prior to its selection. To avoid such requirement, (21) or equivalently (23), which is an approximation of (19), then selects $\mathcal{W}$ such that $L_{\overline{\mathcal{W}}}$ is minimized. Though there is no guarantee that such a reconfiguration policy will lead to minimal performance loss, the numerical analysis presented in the sequel demonstrates that such a compromise does yield satisfactory control performance. Moreover, since the computation of L_n only depends on the current control input sequence, (23) can be solved by computing (22) for each $n \in \mathcal{N}$ and then picking the d number of components whose corresponding L_n values are the largest.*

Putting everything together, the proposed loss-based reconfigurable MPC, or loss-based ReMPC, is summarized in Algorithm 2. Note that Algorithm 2 summarized the required computation for each time step.

Algorithm 2: Loss-based ReMPC

Input: M_k^n, t, $\bar{u}^n$, d, $\tilde{x}^n$, dynamics (1)
Output: u, $\bar{u}^n$

if $t = 0$ **then**
  $\mathcal{W} \leftarrow \mathcal{N}$;
else
  for $n = 1$ *to* N **do**
    $\{\bar{x}_k^n\} \leftarrow$ Integrating (1) using $\bar{u}^n$ and $\tilde{x}^n$;
    $L_n \leftarrow$ (22);
  end
  Rank L_n from highest to the lowest;
  $\mathcal{W} \leftarrow$ (23); % Choose $\mathcal{W}$ such that corresponding components have largest L_n value
end
Solve MPC($\mathcal{W}$) as formulated by (5);
for $n \in \mathcal{W}$ **do**
  $\hat{u}^n \leftarrow$ Solution of (5);
  $\bar{u}^n \leftarrow \hat{u}^n$;
end
for $n \in \overline{\mathcal{W}}$ **do**
  $\hat{u}^n \leftarrow \bar{u}^n$;
end
$u \leftarrow \{\hat{u}_0^n\}$

IV. Practical Applications

This section presents two practical applications of the proposed ReMPC to demonstrate its effectiveness of reducing computation requirement while at the same time maintaining control performance.

A. Battery Cell Balancing Control

Consider the battery cell balancing system in [41], which consists of N battery cells connected in series. The system can be modeled as a distributed system with the dynamics of cell n (or component n) being modeled as [42]:

$$s_{k+1}^n = s_k^n - \frac{I_k + u_k^n}{3600 C^n} T_s \tag{24a}$$

$$v_{k+1}^n = v_k^n - \frac{T_s}{R_p^n C_p^n} v_k^n + \frac{I_k + u_k^n}{C_p^n} T_s \tag{24b}$$

$$v_{o,k}^n = V_{oc,k}^n - v_{k+1}^n - (I_k + u_k^n) R_o^n, \tag{24c}$$

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 [41], [43]–[47] 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

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

The constraints are defined as

$$\mathcal{U} = \left\{ u_k^1, \ldots, u_k^N \mid \sum_n u_k^n = 0, |u_k^n - u_{k-1}^n| \leq \delta_u \right\}$$

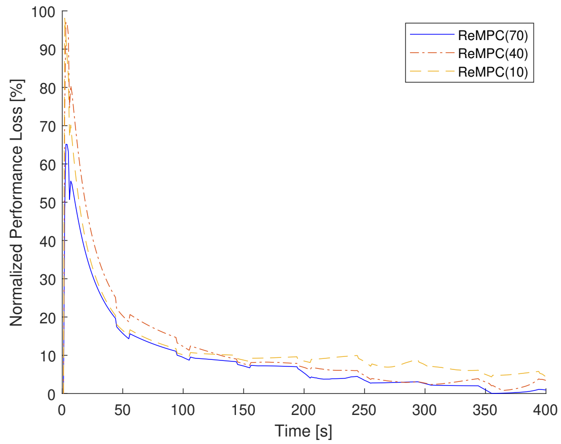

Fig. 1. Normalized performance loss L for different d for battery cell balancing control example.

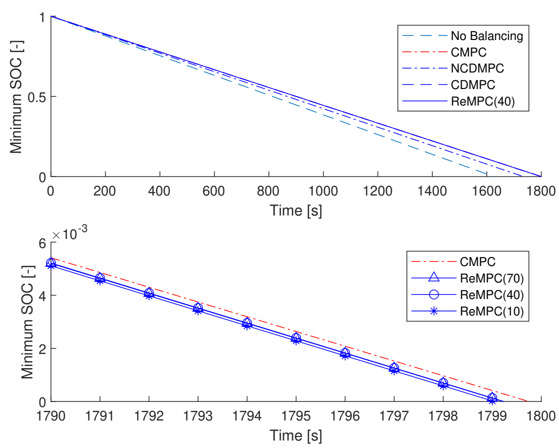

Fig. 2. *Top*: Comparison of minimum SOC for different controllers. *Bottom*: Comparison of minimum SOC for ReMPC with different d.

$$\mathcal{Y} = \left\{ y_k^1, \ldots, y_k^N \mid s_k^n \geq 0, v_{o,k}^n \geq v_{\min} \right\},$$

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 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 [16], 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.

TABLE I
COMPARISON OF BATTERY BALANCING RESULTS

Controller	Range [s]	Extension	Relative Solver Time
No balancing	1,626	-	-
CMPC	1,799	10.701%	100%
ReMPC(10)	1,799	10.701%	4.77%
ReMPC(40)	1,799	10.701%	20.00%
ReMPC(70)	1,799	10.701%	53.57%
NCDMPC	1,735	6.704%	24.95%
CDMPC	1,799	10.701%	48.25%

TABLE II
PARAMETERS FOR MULTI-VEHICLE FORMATION CONTROL

$u_{\min}$	-1	$u_{\max}$	1
$du_{\min}$	-0.05	$du_{\max}$	0.05
$d_{x,ref}^1$	0.1	$d_{y,ref}^1$	-0.2
$d_{x,ref}^2$	-0.1	$d_{y,ref}^2$	-0.2
$d_{x,ref}^3$	-0.1	$d_{y,ref}^3$	0.2
$d_{x,ref}^4$	0.1	$d_{y,ref}^4$	0.2
Q_u	$[1, 0; 01]$	Qy	$\text{diag}([1, 1, 1, 0.7])$
x_0^1	$[0.4; -0.1; 0.4; 0]$	x_0^2	$[0.5; -0.1; 0.1; 0]$
x_0^3	$[0.4; -0.1; -0.4; 0]$	x_0^4	$[0.3; -0.1; -0.1; 0]$

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 compared to CMPC. In particular, the last column of Table I summarizes the relative solver time of each controller, where the average computation time to solve an CMPC instance is used as baseline, i.e., denoted as 100% in Table I. As can be seen, solving the OCP for ReMPC(10) requires only 4.77% of solver time compared to CMPC, 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.

B. *Multi-Vehicle Formation Control*

Consider the multi-vehicle formation control problem studied in [40], [48], [49], where there are a total number of $N = 4$ vehicles. The dynamics of vehicle n can be modeled as

$$p_{x,k+1}^n = p_{x,k}^n + T_s v_{x,k}^n + \frac{T_s^2}{2m^n} u_{x,k}^n \tag{25a}$$

$$v_{x,k+1}^n = v_{x,k}^n + \frac{T_s}{m^n} u_{x,k}^n \tag{25b}$$

$$p_{y,k+1}^n = p_{y,k}^n + T_s v_{y,k}^n + \frac{T_s^2}{2m^n} u_{y,k}^n \tag{25c}$$

$$v_{y,k+1}^n = v_{y,k}^n + \frac{T_s}{m^n} u_{y,k}^n, \tag{25d}$$

where $p_{x,k}^n$ and $v_{x,k}^n$ denote the horizontal position and velocity, respectively, and $p_{y,k}^n$ and $v_{y,k}^n$ denote the vertical position and velocity, respectively. Control input are the horizontal force $u_{x,k}^n$ and vertical force $u_{y,k}^n$. Sampling time $T = 0.1s$ and the mass of vehicle n is $m^n = 0.1kg$.

Define the following variables for inter-vehicle distance:

$$\begin{aligned}
d_{x,k}^1 &= p_{x,k}^2 - p_{x,k}^1, \quad & d_{y,k}^1 &= p_{y,k}^2 - p_{y,k}^1 \\
d_{x,k}^2 &= p_{x,k}^3 - p_{x,k}^2, \quad & d_{y,k}^2 &= p_{y,k}^3 - p_{y,k}^2 \\
d_{x,k}^3 &= p_{x,k}^4 - p_{x,k}^3, \quad & d_{y,k}^3 &= p_{y,k}^4 - p_{y,k}^3 \\
d_{x,k}^4 &= p_{x,k}^1 - p_{x,k}^4, \quad & d_{y,k}^4 &= p_{y,k}^1 - p_{y,k}^4,
\end{aligned}$$

and the output for vehicle n can be defined as

$$y_k^n = \begin{bmatrix} d_{x,k}^n \\ v_{x,k}^n \\ d_{y,k}^n \\ v_{y,k}^n \end{bmatrix} - \begin{bmatrix} d_{x,ref}^n \\ 0 \\ d_{y,ref}^n \\ 0 \end{bmatrix}.$$

In this case, regulating y_k^n towards 0 will effectively regulate both horizontal and vertical velocities to 0 and at the same time maintain the desired inter-vehicle distances as specified by $d_{x,ref}^n$ and $d_{y,ref}^n$. The constraints are defined as

$$\begin{aligned}
u_{\min} &\le u_{x,k}^n \le u_{\max} \\
u_{\min} &\le u_{y,k}^n \le u_{\max} \\
du_{\min} &\le u_{x,k+1}^n - u_{x,k}^n \le du_{\max} \\
du_{\min} &\le u_{y,k+1}^n - u_{y,k}^n \le du_{\max}.
\end{aligned}$$

The proposed ReMPC with $d = 2$ is implemented, i.e., only 2 vehicles are optimized at each given time step, and compared to the performances of (i) a centralized MPC (CMPC) where all vehicles are optimized at every single time step by solving (3) and (ii) a cooperative distributed MPC (DMPC) as discussed in [16]. The parameters used in the simulation are listed in Table II, where x_0^n is the initial condition for vehicle n. The results are plotted in Figs. 3 and 4. Comparing ReMPC and CMPC, it can be seen that the closed-loop system with ReMPC can effectively converge with a settling time comparable to CMPC, despite the fact that there are small overshoots during the transient. It is worth noting that ReMPC and DMPC experience similar transient behavior. In terms of computation time, since ReMPC solves smaller OCP at eath time step, it only consumes 22.71% computation time compared to CMPC and 55.78% computation compared to DMPC, making it suitable for real-time embedded control. It is worth pointing out that, ReMPC can result in a slightly degraded transient behavior compared to CMPC. However, such slight degradation is offset by the computation saving, as discussed above.

To analyze the scalability of the proposed ReMPC framework, the number of vehicles is extended to 10, 20, 30, and 40, and for each case, the centralized MPC, distributed MPC, and the proposed ReMPC are simulated. In addition, two different values for d are simulated, namely, $d_1 = \lfloor 0.25N \rfloor$, $d_2 = \lfloor 0.75N \rfloor$, where recall that N is the number of vehicles. Table III lists the terminal cost of each controller, where all the controllers can effectively stabilize the system with a terminal cost of essentially 0. Using the computational time required by centralized MPC for the case of $N = 10$ as a baseline, the relative computational time required for each controller is listed in Table IV, where the unit is %. Note that the computational time required by CMPC grows exponentially as N increases. Both ReMPC and DMPC require computational time that is linear with respect to the number of vehicles. Depending on the value of d, ReMPC requires similar or even less computational time than DMPC, while at the same time achieving comparable control performance to CMPC and DMPC, as demonstrated in Table III.

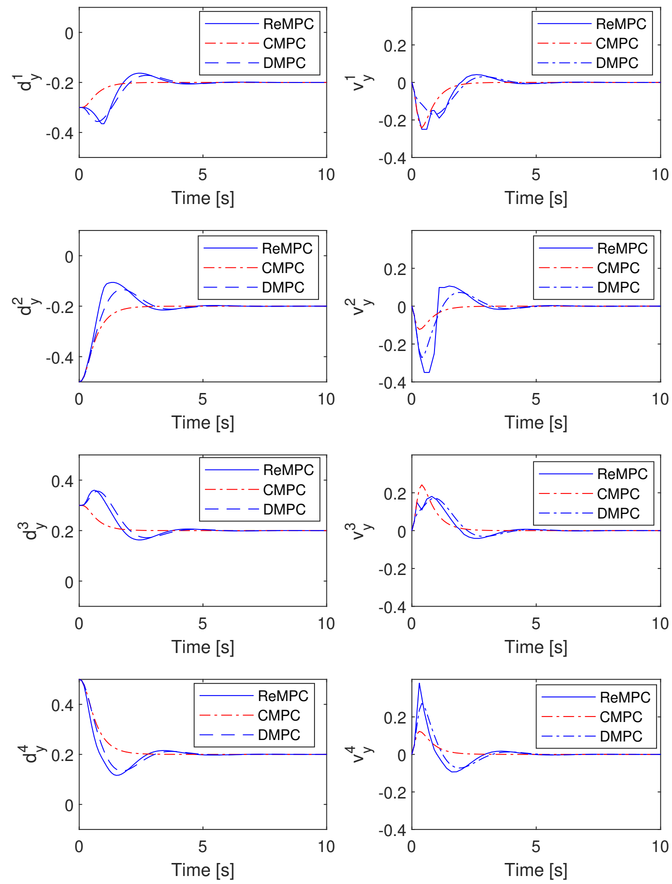

Fig. 3. Lateral formation and lateral speed of each vehicle.

Note that in both battery balancing control and multi-vehicle formation control examples, d is fixed to a predefined value for numerical study. In the future, we will explore the dynamic selection of d so that a larger problem is formulated during the transient conditions while a smaller value for d can be used for steady-state conditions.

V. Conclusion

This paper presents a new reconfigurable model predictive control (MPC) framework, or ReMPC, for large scale

TABLE III
TERMINAL COST FOR FORMATION CONTROL

N	10	20	30	40
CMPC	1.06×10^{-6}	3.18×10^{-4}	6.42×10^{-4}	7.13×10^{-4}
ReMPC(d_1)	1.17×10^{-3}	3.03×10^{-4}	2.34×10^{-3}	3.08×10^{-3}
ReMPC(d_2)	1.73×10^{-7}	2.93×10^{-4}	2.09×10^{-3}	3.05×10^{-3}
DMPC	1.85×10^{-7}	2.94×10^{-4}	2.12×10^{-3}	3.04×10^{-3}

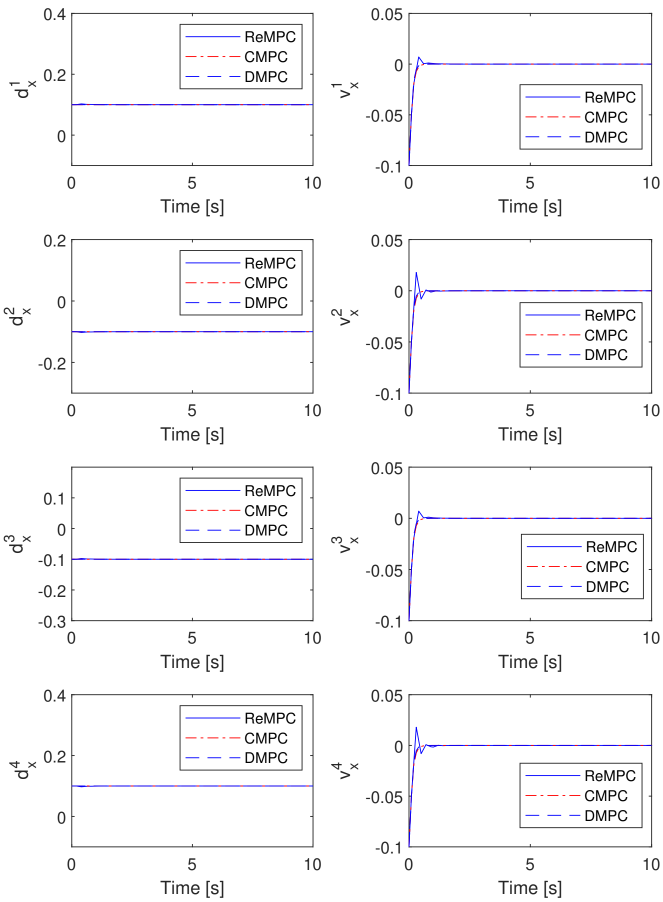

Fig. 4. Longitudinal formation and longitudinal speed of each vehicle.

TABLE IV
RELATIVE COMPUTATIONAL TIME FOR FORMATION CONTROL. UNIT [%].

N	10	20	30	40
CMPC	100	724.7	2558	64040
ReMPC(d_1)	1.73	3.36	8.39	20.82
ReMPC(d_2)	7.33	11.13	24.41	45.28
DMPC	4.49	9.56	11.11	20.72

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 control inputs is dynamically selected to be optimized by MPC, while the previous optimal solution is applied to the remaining control inputs. A theoretical upper bound on performance loss is then derived to guarantee the worst-case performance, which is later used to guide the selection of control inputs to be optimized. Advantage over conventional centralized MPC and distributed MPC is clearly validated through two numerical examples, one on the battery cell-to-cell balancing control with 100 cells and the other on multi-vehicle formation control. It is demonstrated that the proposed ReMPC can achieve better control performance compared to distributed MPC while requiring less computation time compared to centralized MPC. Future work includes (i) the calculation of d to balance performance loss and computation time explicitly, (ii) further improvement for a tighter upperbound on the performance loss by considering the solution cone for MPC($\mathcal{W}$), (iii) analysis on recursive feasibility without Assumption 2 and stability analysis, (iv) the design of state observer and the inclusion of state estimation covariance as part of the selection criteria to determine $\mathcal{W}$, and (v) demonstration of the proposed ReMPC in other application domains such as energy systems [50].

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] Y. Yang, H.-G. Yeh, and R. Nguyen, "A robust model predictive control-based scheduling approach for electric vehicle charging with photovoltaic systems," *IEEE Systems Journal*, 2022.

[5] M. R. C. Qazani, H. Asadi, and S. Nahavandi, "A motion cueing algorithm based on model predictive control using terminal conditions in urban driving scenario," *IEEE Systems Journal*, vol. 15, no. 1, pp. 445–453, 2020.

[6] T. A. Johansen, "Toward dependable embedded model predictive control," *IEEE Systems Journal*, vol. 11, no. 2, pp. 1208–1219, 2014.

[7] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[8] J. Chen and Z. Yi, "Comparison of event-triggered model predictive control for autonomous vehicle path tracking," in *IEEE Conference on Control Technology and Applications*, San Diego, CA, August 8–11, 2021.

[9] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[10] E. Henriksson, D. E. Quevedo, E. G. Peters, H. Sandberg, and K. H. Johansson, "Multiple-loop self-triggered model predictive control for network scheduling and control," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2167–2181, 2015.

[11] F. D. Brunner, W. Heemels, and F. Allgöwer, "Robust event-triggered MPC with guaranteed asymptotic bound and average sampling rate," *IEEE Transactions on Automatic Control*, vol. 62, no. 11, pp. 5694–5709, 2017.

[12] T. Marcucci and R. Tedrake, "Warm start of mixed-integer programs for model predictive control of hybrid systems," *IEEE Transactions on Automatic Control*, vol. 66, no. 6, pp. 2433–2448, June 2021.

[13] L. Schwenkel, M. Gharbi, S. Trimpe, and C. Ebenbauer, "Online learning with stability guarantees: A memory-based warm starting for real-time MPC," *Automatica*, vol. 122, p. 109247, 2020.

[14] M. N. Zeilinger, C. N. Jones, and M. Morari, "Real-time suboptimal model predictive control using a combination of explicit MPC and online optimization," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1524–1534, 2011.

[15] D. Liao-McPherson, M. M. Nicotra, A. L. Dontchev, I. V. Kolmanovsky, and V. Veliov, "Sensitivity-based warmstarting for nonlinear model predictive control with polyhedral state and control constraints," *IEEE Transactions on Automatic Control*, 2019.

[16] 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.

[17] M. Kordestani, A. A. Safavi, N. Sharafi, and M. Saif, "Novel multiagent model-predictive control performance indices for monitoring of a large-scale distributed water system," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1286–1294, 2016.

[18] 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.

[19] 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.

[20] A. Ferrara, A. N. Oleari, S. Sacone, and S. Siri, "Freeways as systems of systems: A distributed model predictive control scheme," *IEEE Systems Journal*, vol. 9, no. 1, pp. 312–323, 2014.

[21] 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.

[22] 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.

[23] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright, "Distributed mpc strategies with application to power system automatic generation control," *IEEE transactions on control systems technology*, vol. 16, no. 6, pp. 1192–1206, 2008.

[24] J. Chen, X. Meng, and Z. Li, "Reinforcement learning-based event-triggered model predictive control for autonomous vehicle path following," in *2022 American Control Conference*, Atlanta, GA, June 8–10, 2022.

[25] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, Seoul, Korea, 2015, pp. 1094–1099.

[26] M. Ostadijafari and A. Dubey, "Tube-based model predictive controller for building's heating ventilation and air conditioning (hvac) system," *IEEE Systems Journal*, vol. 15, no. 4, pp. 4735–4744, 2020.

[27] A. Dutta, S. Ganguly, and C. Kumar, "Coordinated volt/var control of PV and EV interfaced active distribution networks based on dual-stage model predictive control," *IEEE Systems Journal*, vol. 16, no. 3, pp. 4291–4300, 2021.

[28] K.-V. Ling, J. Maciejowski, J. Guo, and E. Siva, "Channel-hopping model predictive control," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 11 417–11 422, 2011.

[29] Z. Zhou, C. Rother, and J. Chen, "Event-triggered model predictive control for autonomous vehicle path tracking: Validation using CARLA simulator," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 6, pp. 3547–3555, June 2023.

[30] H. Li and Y. Shi, "Event-triggered robust model predictive control of continuous-time nonlinear systems," *Automatica*, vol. 50, no. 5, pp. 1507–1513, 2014.

[31] H. Li, W. Yan, and Y. Shi, "Triggering and control codesign in self-triggered model predictive control of constrained systems: With guaranteed performance," *IEEE Transactions on Automatic Control*, vol. 63, no. 11, pp. 4008–4015, 2018.

[32] C. Liu, H. Li, Y. Shi, and D. Xu, "Codesign of event trigger and feedback policy in robust model predictive control," *IEEE Transactions on Automatic Control*, vol. 65, no. 1, pp. 302–309, 2019.

[33] K. Hashimoto, S. Adachi, and D. V. Dimarogonas, "Self-triggered model predictive control for nonlinear input-affine dynamical systems via adaptive control samples selection," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 177–189, 2016.

[34] F. Dang, D. Chen, J. Chen, and Z. Li, "Event-triggered model predictive control with deep reinforcement learning," *IEEE Transactions on Intelligent Vehicles*, accepted for Publication, October 2023.

[35] F. D. Brunner, M. Heemels, and F. Allgöwer, "Robust self-triggered MPC for constrained linear systems: A tube-based approach," *Automatica*, vol. 72, pp. 73–83, 2016.

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

[37] 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, July 2021.

[38] 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.

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

[40] Y. Zou, X. Su, S. Li, Y. Niu, and D. Li, "Event-triggered distributed predictive control for asynchronous coordination of multi-agent systems," *Automatica*, vol. 99, pp. 92–98, 2019.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] J. Chen, A. Behal, and C. Li, "Active battery cell balancing by real time model predictive control for extending electric vehicle driving range," *IEEE Transactions on Automation Science and Engineering*, accepted June 2023.

[48] C. Wang and C.-J. Ong, "Distributed model predictive control of dynamically decoupled systems with coupled cost," *Automatica*, vol. 46, no. 12, pp. 2053–2058, 2010.

[49] B. Ding, L. Xie, and W. Cai, "Distributed model predictive control for constrained linear systems," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 11, pp. 1285–1298, 2010.

[50] K. Deng, Y. Sun, S. Li, Y. Lu, J. Brouwer, P. G. Mehta, M. Zhou, and A. Chakraborty, "Model predictive control of central chiller plant with thermal energy storage via dynamic programming and mixed-integer linear programming," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 565–579, April 2015.