2026 American Control Conference (ACC)
May 27-29, 2026. New Orleans, Louisiana, USA

Asynchronous Model Predictive Control for Large Interconnected Systems with External Disturbance

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

***Abstract*— For large interconnected systems, centralized model predictive control (MPC) may be computationally intractable while distributed MPC may not achieve global optimum. To fill this gap, a new MPC framework, named asynchronous MPC, has been proposed in literature, where only a subset of subsystems is selected for optimization at each time step. For the remaining subsystems, the previously optimized control sequence is shifted and reused. Such an approach can balance computational load and control performance, and can be viewed as a framework in between centralized MPC and distributed MPC. However, prior work assumes each subsystem to be deterministic and does not consider the impacts of external disturbance. Moreover, feasibility and stability analysis for asynchronous MPC has not been studied. This paper addresses these limitations by (i) incorporating the prediction error introduced by external disturbance into reconfiguration policy, (ii) providing a stability condition for the closed-loop system, and (iii) establishing the recursive feasibility. Numerical results using battery cell-to-cell balancing control confirms the stability of the closed-loop system and demonstrates over 86% computation reduction.**

I. Introduction

Model predictive control (MPC) has been extensively studied for large systems [1]–[6]. Since centralized MPC suffers the curse of dimensionality, it can be computationally intractable to implement centralized MPC for large interconnected systems. To reduce computation, distributed MPC are often used [7]–[9], which can, in general, only achieve sub-optimal control performance [7]. To balance computational load and control performance, event-triggered MPC has been proposed in literature [10]–[13]. In event-triggered MPC framework, an optimal control problem (OCP) optimizing all control inputs is formulated and solved only when an event is triggered. In other words, control inputs are optimized aperiodically but synchronously in event-triggered MPC. By reducing the number of online optimization problems, event-triggered MPC can effectively reduce the “average” computation time. However, the “worst-case” computation time remains the same, since when an event is triggered, the OCP still consists of a large number of optimization variables.

To reduce both average and worst-case computation time, a new MPC framework, named asynchronous MPC (AsynMPC) is proposed in [14], where the structure of the OCP is reconfigured at each control loop. In other words, a (time-varying) subset of control inputs are optimized at each time step to reduce the computation time while keeping the closed-loop control performance comparable to a centralized MPC. Compared to event-triggered MPC, the optimization of control inputs is both aperiodic and asynchronous in AsynMPC, and hence can reduce both average and worst-case computation time.

However, despite the excellent performance demonstrated in [14], several issues remain unaddressed. *Firstly*, the interconnected systems considered in [14] are assumed to be disturbance free, making the reconfiguration approach not robustness to external disturbance. *Secondly*, a feasible solution is assumed to exist for each control loop. In other words, the recursive feasibility is not analyzed in the setting of [14], which limits the applicability of AsynMPC. *Lastly*, while the worst-case performance loss is analyzed in [14], the stability analysis has not been reported. To address these limitations, the main contributions of this paper is to study AsynMPC for large interconnected systems with external disturbance, to provide a stability condition for the closed-loop system with AsynMPC, and to analyze the recursive feasibility of AsynMPC under the assumption that the feasible region is a closed and convex polyhedron.

The rest of this paper is organized as follows. Section II reviews the AsynMPC framework studied in [14] and proposes a new reconfiguration policy that considers external disturbance. Section III analyzes the recursive feasibility and stability of AsynMPC. Numerical simulation results on battery cell-to-cell balancing control of 100 connected cells are presented in Section IV, and 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 and use $\|\cdot\|_Q$ to denote the Q-weighted norm of a vector with Q being symmetric positive semi-definite. In other words,

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

Moreover, we have

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

Given a square matrix A, the Euclidean norm of A is defined as $\|A\| = \sqrt{\lambda(A^T A)}$ where $\lambda(A^T A)$ is the largest eigenvalue of $A^T A$.

This work was supported in part by National Science Foundation through Award #2237317. Jun Chen is the corresponding author.

Jun Chen is with the Department of Electrical and Computer Engineering, Oakland University, Rochester, MI 48374 USA (e-mail: junchen@oakland.edu). Chong Li is with OORT and the Department of Electrical Engineering, Columbia University, NYC, 10027, USA (email: cl3607@columbia.edu).

II. Asynchronous Model Predictive Control

A. MPC for Interconnected Systems

Consider an interconnected system with N linear discrete-time subsystems, and the nth subsystem has the following dynamics:

$$x_{n,t+1} = A_n x_{n,t} + B_n u_{n,t} + w_{n,t}, \qquad n \in \mathcal{N} \tag{1}$$

where $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of all subsystems, $x_n \in \mathbb{R}^{n_x}$, $u_n \in \mathbb{R}^{n_u}$, and $w_n \in \mathbb{R}^{n_x}$ are the states, inputs, and external disturbances for nth subsystem, and A_n and B_n are system matrices with proper dimension. The external disturbance w_n is assumed to be bounded by $\|w_n\| \leq \bar{w}_n$.

Denote the input and state variables of the system as

$$\mathbf{u}_t = \begin{bmatrix} u_{1,t}^T & u_{2,t}^T & \ldots & u_{N,t}^T \end{bmatrix}^T$$
$$\mathbf{x}_t = \begin{bmatrix} x_{1,t}^T & x_{2,t}^T & \ldots & x_{N,t}^T \end{bmatrix}^T.$$

The subsystem inputs are assumed to be coupled through constraints, as follows:

$$\mathbf{u}_t \in \mathbb{U} \subseteq \mathbb{R}^{Nn_u},$$

where $\mathbb{U}$ is a closed and convex polyhedron satisfying $\mathbf{0} \in \mathbb{U}$.

Remark 1: The interconnected system (1) assumes the subsystems have decoupled dynamics and coupled input constraints. Such a setting can find several applications. For example, the multi-agent systems considered in [15] can be modeled using (1). Vehicle platooning, when ignored the aerodynamics between vehicles [16], can also be modeled using (1). Moreover, serial-connected battery cells as studied in [17], [18] also satisfy (1) with decoupled dynamics and coupled input constraints. ■

The nominal model for the nth subsystem is given by

$$x_{n,t+1} = A_n x_{n,t} + B_n u_{n,t}, \qquad n \in \mathcal{N}. \tag{2}$$

We made the following assumptions about (2), similar to those in [15], [19], [20].

Assumption 1: Given symmetric positive definite matrices Q_n and R_n, there exists a constant $\epsilon_n > 0$, a matrix $P_n > 0$, and a local state feedback gain K_n such that:

1) $\phi_n = \{x_n | \, \|x_n\|_{P_n}^2 \leq \epsilon_n^2\}$ is positive invariant for subsystem n. In other words, $\forall x_{n,t} \in \phi_n, (A_n + B_n K_n) x_{n,t} \in \phi_n$.
2) $Q_n + K_n^T R_n K_n + (A_n + B_n K_n)^T P_n (A_n + B_n K_n) - P_n \leq 0$.
3) $\forall \mathbf{x}_t = \begin{bmatrix} x_{1,t}^T & x_{2,t}^T & \ldots & x_{N,t}^T \end{bmatrix}^T$ with $x_{n,t} \in \phi_n$, $\mathbf{u}_t = \begin{bmatrix} u_{1,t}^T & u_{2,t}^T & \ldots & u_{N,t}^T \end{bmatrix}^T$ with $u_{n,t} = K_n x_{n,t}$ is feasible, i.e., $\mathbf{u}_t \in \mathbb{U}$.

At each time step t, given current state estimate $\hat{\mathbf{x}}_t$, a centralized MPC solves the following optimal control problem (OCP) over a prediction horizon p, where the control sequence is defined as $\mathbf{u}_{t,\ldots,t+p-1|t} = \begin{bmatrix} \mathbf{u}_{t|t}^T & \mathbf{u}_{t+1|t}^T & \ldots & \mathbf{u}_{t+p-1|t}^T \end{bmatrix}^T$:

$$\min_{\mathbf{u}_{t,\ldots,t+p-1|t}} \quad J = \sum_{k=0}^{p-1} \ell(\mathbf{x}_{t+k|t}, \mathbf{u}_{t+k|t}) + V_f(\mathbf{x}_{t+p|t}) \tag{3a}$$
$$\text{s.t.} \quad (2) \text{ with initial condition } \hat{\mathbf{x}}_t, \tag{3b}$$
$$\mathbf{u}_{t+k|t} \in \mathbb{U}, \ \forall k = 0, 1, \ldots, p-1, \tag{3c}$$
$$x_{n,t+p|t} \in \phi_n, \ \forall n \in \mathcal{N}, \tag{3d}$$

where the stage and terminal costs are strictly convex quadratic

$$\ell(\mathbf{x}, \mathbf{u}) = ||\mathbf{x}||_{\mathbf{Q}}^2 + ||\mathbf{u}||_{\mathbf{R}}^2$$
$$V_f(\mathbf{x}) = ||\mathbf{x}||_{\mathbf{P}}^2$$

with the constant weighting matrices

$$\mathbf{Q} = \begin{bmatrix} Q_1 & 0 & \cdots & 0 \\ 0 & Q_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & Q_N \end{bmatrix}$$
$$\mathbf{R} = \begin{bmatrix} R_1 & 0 & \cdots & 0 \\ 0 & R_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & R_N \end{bmatrix}$$
$$\mathbf{P} = \begin{bmatrix} P_1 & 0 & \cdots & 0 \\ 0 & P_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & P_N \end{bmatrix}.$$

Note that the weight matrices Q_n, R_n and P_n are all assumed to be symmetric positive definite satisfying Assumption 1. 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 computationally intractable (even for small prediction horizon p) due to the high computational requirement. To address this issue, an asynchronous MPC (AsynMPC) framework is proposed below for the nominal systems (2), where a subset of subsystems is dynamically selected to form the OCP, while for the remaining subsystems, previous optimal solution is applied as control inputs.

B. Asynchronous MPC

Given a subset $\mathcal{W}_t \subseteq \mathcal{N}$, denote its complementary set as $\overline{\mathcal{W}}_t = \mathcal{N} - \mathcal{W}_t$, i.e., $\mathcal{W}_t \cup \overline{\mathcal{W}}_t = \mathcal{N}$ and $\mathcal{W}_t \cap \overline{\mathcal{W}}_t = \emptyset$. The proposed AsynMPC formulates a reduced size OCP that only includes subsystems in $\mathcal{W}_t$. Note that for each time step, the set $\mathcal{W}_t$ of subsystems for optimization is varying. Here we use the subscript t to denote the fact that both the content of $\mathcal{W}_t$ and its cardinality are time-varying. For subsystems that are in $\overline{\mathcal{W}}_t$, their control inputs are not included in the reduced size OCP and need to be obtained somewhere else. Let d_n be the time instance that subsystem n is selected for optimization, i.e., $n \in \mathcal{W}_{d_n}$, and denote the corresponding optimal control sequence for subsystem n as $u^*_{n,d_n+k|d_n}$, $k = 0, \ldots, p-1$. Also denote d'_n as the next time instance that subsystem n is included for optimization. In other words, $n \notin \mathcal{W}_{d_n+m}$ for $d_n < d_n + m < d'_n$ and $n \in \mathcal{W}_{d'_n}$. Then the following candidate control sequence $\bar{u}_n$ is constructed for subsystem n for $m = 1, \ldots, d'_n - d_n - 1$.

$$\bar{u}_{n,d_n+l|d_n+m} = \begin{cases} u^*_{n,d_n+l|d_n} & l = m, \ldots, p-1 \\ K_n x^*_{n,d_n+l|d_n} & l = p, \ldots, m+p-1, \end{cases} \tag{4}$$

where $x^*_{n,d_n+l|d_n}$ is the predicted state at time $d_n + l$ based on the control sequence (4). In other words, the solution obtained at time step d_n for subsystem n is shifted to obtained a new control sequence for time instances before d'_n. And when this solution is used up, local state feedback gain K_n is used to construct the remainder of the new control sequence.

At time step t, denote the size of $\mathcal{W}_t$ as τ and then denote $\mathcal{W}_t = \{i_1, i_2, \ldots, i_\tau\}$, where $1 \le i_1 < i_2 < \cdots < i_\tau \le N$. Furthermore, define the index function κ_t as $\kappa_t(n) = j$ if $n = i_j \in \mathcal{W}_t$ and otherwise $\kappa_t(n)$ is undefined. Similarly, denote the size of $\overline{\mathcal{W}}_t$ as $\bar{\tau}$ and then denote $\overline{\mathcal{W}}_t = \{\bar{i}_1, \bar{i}_2, \ldots, \bar{i}_{\bar{\tau}}\}$, where $1 \le \bar{i}_1 < \bar{i}_2 < \cdots < \bar{i}_{\bar{\tau}} \le N$. Furthermore, define the index function $\bar{\kappa}_t$ as $\bar{\kappa}_t(n) = j$ if $n = \bar{i}_j \in \overline{\mathcal{W}}_t$ and otherwise $\bar{\kappa}_t(n)$ is undefined.

The input and state variables associated with $\mathcal{W}_t$ can now be denoted as

$$\tilde{\mathbf{u}}_{t+k|t} = \begin{bmatrix} u^T_{i_1,t+k|t} & u^T_{i_2,t+k|t} & \cdots & u^T_{i_\tau,t+k|t} \end{bmatrix}^T$$
$$\tilde{\mathbf{x}}_{t+k|t} = \begin{bmatrix} x^T_{i_1,t+k|t} & x^T_{i_2,t+k|t} & \cdots & x^T_{i_\tau,t+k|t} \end{bmatrix}^T,$$

and the input variable associated with $\overline{\mathcal{W}}_t$ can now be denoted as

$$\bar{\mathbf{u}}_{t+k|t} = \begin{bmatrix} \bar{u}^T_{\bar{i}_1,t+k|t} & \bar{u}^T_{\bar{i}_2,t+k|t} & \cdots & \bar{u}^T_{\bar{i}_{\bar{\tau}},t+k|t} \end{bmatrix}^T,$$

where $\bar{u}_{n,t+k|t}$ for $n \in \overline{\mathcal{W}}_t$ is defined in (4).

Given $\mathcal{W}_t$ and $\bar{\mathbf{u}}_{t+k|t}$, the inputs of each subsystem in $\mathcal{W}_t$ are then coupled through the following constraints:

$$\tilde{\mathbf{u}}_{t+k|t} \in \tilde{\mathbb{U}}_{t+k|t} \subseteq \mathbb{R}^{\tau n_u}.$$

Here $\tilde{\mathbb{U}}_{t+k|t}$ is given as follows.

$$\tilde{\mathbb{U}}_{t+k|t} = \{\tilde{\mathbf{u}}_{t+k|t} | \texttt{Assemble}(\mathcal{W}_t, \tilde{\mathbf{u}}_{t+k|t}, \bar{\mathbf{u}}_{t+k|t}) \in \mathbb{U}\}, \tag{5}$$

where the function $\texttt{Assemble}(\mathcal{W}_t, \tilde{\mathbf{u}}_{t+k|t}, \bar{\mathbf{u}}_{t+k|t}) = \begin{bmatrix} u^T_{1,t+k|t} & u^T_{2,t+k|t} & \cdots & u^T_{N,t+k|t} \end{bmatrix}^T$ with

$$u_{n,t+k|t} = \begin{cases} \tilde{\mathbf{u}}_{t+k|t}(\kappa_t(n)) & \text{if } n \in \mathcal{W}_t \\ \bar{\mathbf{u}}_{t+k|t}(\bar{\kappa}_t(n)) & \text{if } n \notin \mathcal{W}_t. \end{cases}$$

Note that here $\tilde{\mathbf{u}}_t(i)$ and $\bar{\mathbf{u}}_t(i)$ denote the ith block of the vector $\tilde{\mathbf{u}}$ and $\bar{\mathbf{u}}$, respectively.

Now we are ready to formulate a reduced size OCP that only includes subsystems in $\mathcal{W}_t$. At each time step t, given current state estimate $\hat{\tilde{\mathbf{x}}}_t$, the following reduced size OCP is formulated, where the control sequence is defined as $\tilde{\mathbf{u}}_{t,\ldots,t+p-1|t} = \begin{bmatrix} \tilde{\mathbf{u}}^T_{t|t} & \tilde{\mathbf{u}}^T_{t+1|t} & \cdots & \tilde{\mathbf{u}}^T_{t+p-1|t} \end{bmatrix}^T$:

$$\min_{\tilde{\mathbf{u}}_{t,\ldots,t+p-1|t}} \quad \tilde{J} = \sum_{k=0}^{p-1} \tilde{\ell}(\tilde{\mathbf{x}}_{t+k|t}, \tilde{\mathbf{u}}_{t+k|t}) + \tilde{V}_f(\tilde{\mathbf{x}}_{t+p|t}) \tag{6a}$$

$$\text{s.t.} \quad (2) \text{ with initial condition } \hat{\tilde{\mathbf{x}}}_t, \ \forall n \in \mathcal{W}_t \tag{6b}$$
$$\tilde{\mathbf{u}}_{t+k|t} \in \tilde{\mathbb{U}}_{t+k|t}, \ \forall k = 0, 1, \ldots, p-1, \tag{6c}$$
$$x_{n,t+p|t} \in \phi_n, \ \forall n \in \mathcal{W}_t, \tag{6d}$$

where the stage and terminal cost are

$$\tilde{\ell}(\tilde{\mathbf{x}}, \tilde{\mathbf{u}}) = ||\tilde{\mathbf{x}}||^2_{\tilde{\mathbf{Q}}} + ||\tilde{\mathbf{u}}||^2_{\tilde{\mathbf{R}}}$$
$$\tilde{V}_f(\mathbf{x}) = ||\tilde{\mathbf{x}}||^2_{\tilde{\mathbf{P}}},$$

with the weighting matrices $\tilde{\mathbf{Q}}$, $\tilde{\mathbf{R}}$, and $\tilde{\mathbf{P}}$ are all $\tau \times \tau$ block diagonal matrices given as

$$\tilde{\mathbf{Q}} = \begin{bmatrix} Q_{i_1} & 0 & \cdots & 0 \\ 0 & Q_{i_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & Q_{i_\tau} \end{bmatrix}$$
$$\tilde{\mathbf{R}} = \begin{bmatrix} R_{i_1} & 0 & \cdots & 0 \\ 0 & R_{i_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & R_{i_\tau} \end{bmatrix}$$
$$\tilde{\mathbf{P}} = \begin{bmatrix} P_{i_1} & 0 & \cdots & 0 \\ 0 & P_{i_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & P_{i_\tau} \end{bmatrix}.$$

For each control loop, AsynMPC selects a set $\mathcal{W}_t$ of subsystems, solves OCP (6) to obtain the optimal solution $\tilde{\mathbf{u}}^*_{t,\ldots,t+p-1|t} = \begin{bmatrix} (\tilde{\mathbf{u}}^*_{t|t})^T & (\tilde{\mathbf{u}}^*_{t+1|t})^T & \cdots & (\tilde{\mathbf{u}}^*_{t+p-1|t})^T \end{bmatrix}^T$, and assembles the control sequence $\mathbf{u}_{t,\ldots,t+p-1|t} = \begin{bmatrix} \mathbf{u}^T_{t|t} & \mathbf{u}^T_{t+1|t} & \cdots & \mathbf{u}^T_{t+p-1|t} \end{bmatrix}$, where

$$\mathbf{u}_{t+k|t} = \texttt{Assemble}(\mathcal{W}_t, \tilde{\mathbf{u}}^*_{t+k|t}, \bar{\mathbf{u}}_{t+k|t}). \tag{7}$$

For simplicity, we denote the OCP (6) as OCP($\mathcal{W}_t$). Then it is trivial to see that the full size OCP (3) is equivalent to OCP($\mathcal{N}$). Algorithm 1 formally presents the AsynMPC framework. As can be seen, all control inputs are optimized at $t = 1$ (initializatiom) at Line 1 with their solutions stored in memory at Line 2–4, making (4) feasible for all subsequent time steps (see Theorem 1 below). Then for each time step, a new subset $\mathcal{W}_t$ is selected at Line 10 such that for each $n \in \overline{\mathcal{W}}_t$,

$$\begin{aligned} &\sum_{k=0}^{p-2} \left\| A_n^k e_{n,t} \right\|^2_{Q_n} + \sum_{k=0}^{p-1} 2 x_{n,t+k|t-1} Q_n A_n^k e_{n,t} \\ &\quad - \left\| x_{n,t-1|t-1} \right\|^2_{Q_n} - \left\| u_{n,t-1|t-1} \right\|^2_{R_n} \\ &\quad + 2 x^T_{n,t+p-1|t-1} (\bar{R}_n + \bar{P}_n) A_n^{p-1} e_{n,t} \\ &\quad + \left\| A_n^{p-1} e_{n,t} \right\|^2_{P_n} \le 0 \end{aligned} \tag{8a}$$
$$\left\| A_n^{p-1} e_{n,t} \right\|^2_{P_n} \le \epsilon_n \tag{8b}$$

where $e_{n,t} = x_{n,t|t} - x_{n,t|t-1}$, i.e., $e_{n,t}$ is the error between the current state $x_{n,t|t}$ at time t and its prediction $x_{n,t|t-1}$ at time $t-1$ and

$$\bar{R}_n = K_n^T R_n K_n \tag{9a}$$

Algorithm 1: Asynchronous Model Predictive Control

1 Initialize by solving OCP($\mathcal{N}$), i.e., standard OCP (3) for all subsystems, to get optimal solution $\mathbf{u}^*_{0|0}, \mathbf{u}^*_{1|0}, \ldots, \mathbf{u}^*_{p-1|0}$;
2 **for** $n = 1, 2, \ldots, N$ **do**
3   **for** $k = 0, 2, \ldots, p-1$ **do**
4   Memory $\leftarrow \mathbf{u}^*_{k|0}(n)$;
5   **end**
6 **end**
7 Apply first control input $\mathbf{u}_0 = \mathbf{u}^*_{0|0}$ and move to next time step;
8 **while** $t \leq T$ **do**
9   Collect current state estimate $\hat{\mathbf{x}}_t$;
10   Select a new $\mathcal{W}_t \subseteq \mathcal{N}$ such that (8) for each $n \in \overline{\mathcal{W}}_t$ holds;
11   Solve OCP($\mathcal{W}_t$) as formulated by (6) to get optimal solution $\tilde{\mathbf{u}}^*_{t|t}, \tilde{\mathbf{u}}^*_{t+1|t}, \ldots, \tilde{\mathbf{u}}^*_{t+p-1|t}$ for subsystems in $\mathcal{W}_t$;
12   **for** $n \in \mathcal{W}_t$ **do**
13   **for** $k = 0, 2, \ldots, p-1$ **do**
14   Memory $\leftarrow \tilde{\mathbf{u}}^*_{t+k|t}(\kappa_t(n))$;
15   **end**
16   **end**
17   **for** $n \in \overline{\mathcal{W}}_t$ **do**
18   **for** $k = 1, 2, \ldots, p-1$ **do**
19             $\bar{u}_{n,t+k|t} \leftarrow (4)$;
20   Memory $\leftarrow \bar{u}_{n,t+k|t}$;
21   **end**
22   **end**
23   Apply $\mathbf{u}_{t|t}$ according to (7) and move to next time step;
24 **end**

$$\bar{P}_n = (A_n + B_n K_n)^T P_n (A_n + B_n K_n). \tag{9b}$$

The reduced size OCP (6) is then formed, which is also termed as OCP($\mathcal{W}_t$) and solved at Line 11. The latest optimal control sequence for $n \in \mathcal{W}_t$ is then stored in memory at Line 12–16, while for $n \in \overline{\mathcal{W}}_t$, i.e., subsystems not selected for optimization, their previous optimal solutions (as saved in memory) is shifted to obtain $\bar{u}_n$ at Line 17–22. Line 23 applies the first control input for each subsystem and move to the next time step.

Remark 2: Note that OCP($\mathcal{W}_t$) only optimizes control inputs for subsystems in $\mathcal{W}_t$, while for $n \in \overline{\mathcal{W}}_t$, previous optimal solution is used to implement its control, which is also used to form the constraints in (6c). Therefore, the number of optimization variables of OCP($\mathcal{W}_t$) is reduced to $\tau p n_u$. It is also worth noting that the size of $\mathcal{W}_t$, i.e., τ is also time-varying. ■

Remark 3: When the prediction error $e_{n,t}$ is large, τ can be dynamically increased to improve control performance according to (8a). ■

III. Stability and Feasibility of AsynMPC

As can be seen from Line 10 of Algorithm 1, a new subset $\mathcal{W}_t$ is selected at each time step. Since $\mathcal{W}_t$ is time-varying, its selection is crucial to guarantee that all control inputs are updated using measurement feedback, but with heterogeneous and aperiodic sampling time. This section analyzes the stability and feasibility of the proposed AsynMPC.

Let's start with the recursive feasibility. Recall that the feasible region $\mathbb{U}$ is closed and convex polyhedron satisfying $\mathbf{0} \in \mathbb{U}$. The next theorem guarantees the recursive feasibility of AsynMPC.

Theorem 1: Assuming (3) is feasible at time $t = 0$, then constraints (3c) and (3d) are satisfied for all $t > 0$.

Proof: The proof is omitted due to space limitation. ■

Next we show that the closed loop system with AsynMPC is asymptotically stable in the Lyapunov sense. Let's denote J_t as the cost function (3a) under candidate control sequence $\mathbf{u}_{t+k|t}$, $k = 0, \ldots, p-1$, as computed by (7). In the sequel, the difference ΔJ_t between J_t and J_{t-1} is considered.

Theorem 2: For asynchronous MPC presented in Algorithm 1, if we select $\mathcal{W}_t$ such that (8) holds for all $n \in \overline{\mathcal{W}}_t$, then we have $\Delta J_t < 0$.

Proof: The proof is omitted due to space limitation. ■

Remark 4: In the extreme case, none of the subsystem meets condition (8), in which case $\mathcal{W}_t = \mathcal{N}$. According to Assumption 1, the stability condition is still guaranteed in this case. On the other hand, when all the subsystems meet condition (8), case $\mathcal{W}_t = \emptyset$. In this case, none of the subsystems is selected for the reduced size OCP, resulting in skipping the optimization in this instance. Therefore, the case of $\mathcal{W}_t = \emptyset$ results in a mechanism equivalent to event-triggered MPC in the absence of an event [12], [13], [21], [22]. However, when $0 < \tau < N$, AsynMPC still optimizes control inputs for subsystems in $\mathcal{W}_t$, making the optimizations of control inputs both aperiodic and asynchronous. This is different from event-triggered MPC treatment where the optimization of control inputs occurs aperiodically but synchronously. ■

Remark 5: Note that the reconfiguration policy (8) guarantees the close-loop system of (2) is stable. However, it does not guarantee system robustness against the external disturbance w_n. To this regard, one can replace the condition (8b) with the following,

$$\bar{w}_n \leq \frac{1 - \|A\|}{1 - \|A\|^p} \epsilon_n,$$

which guarantees the deviation of states of (1) from the nominal model (2) to be upper bounded by ϵ_n. Therefore, when the system state is sufficiently close to origin, it will remain in the invariant set ϕ_n even under external disturbance w_n. ■

IV. Application to Battery Balancing Control

This section presents the practical application of the proposed AsynMPC to battery cell balancing problem, and demonstrates its effectiveness in reducing computation load while at the same time maintains a comparable control

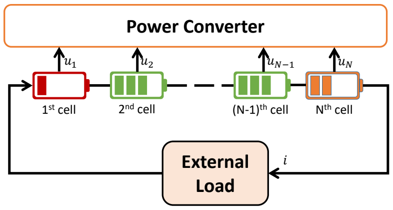

Fig. 1. Illustration of the active battery cell-to-cell balancing.

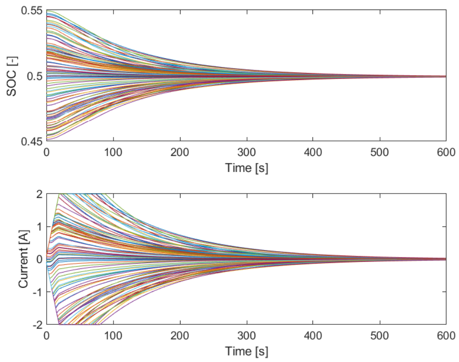

Fig. 2. Simulation results using CMPC. (a) SOC for each cell. (b) Balancing current for each cell.

performance. Consider a battery cell balancing system consisting of N battery cells with serial connection, which is adopted from [17], [18]. See Fig. 1. Each cell's dynamic can be modeled as Coulomb counting as follows [23], [24].

$$s_{n,t+1} = s_{n,t} - \frac{u_{n,t}}{3600C_n}T_s + w_{n,t},$$

where s_n is the state-of-charge (SOC) of cell n, C_n is the cell capacity, and u_n is the balancing current of cell n. Due to variations in manufacturing and operations, each cell can have different capacity C_n, which can lead to different SOC among cells. The objective of battery cell balancing control is then to actively move electric charges from cell to cell through a power converter to maintain a balanced SOC among the battery cells. In this case study, we apply the proposed AsynMPC to active cell balancing control problem with a large number of cells, e.g. $N = 100$.

Assuming that $\bar{s}$ is the balanced SOC level that we want all cells to track, define $x_{n,t} = s_{n,t} - \bar{s}$. Then we have the folowing dynamics for subsystem n:

$$x_{n,t+1} = x_{n,t} - \frac{u_{n,t}}{3600C_n}T_s + w_{n,t}.$$

Furthermore, the input constraint $\mathbb{U}$ is defined as the set of $\begin{bmatrix} u_{1,t}^T & u_{2,t}^T & \dots & u_{N,t}^T \end{bmatrix}^T$ such that $\sum_n u_{n,t} = 0$ and $|u_{n,t} - u_{n,t-1}| \leq \delta_u$. Here the first constraint states that the sum of charge and discharge current should be equal to zero, which is relaxed from the energy conservation constraint. The second constraint limits the change rate of balancing current as specified by the power converter capability.

For simulation purpose, we randomly generate cell parameters C_n to be up to $\pm 10\%$ different from a nominal value, and the initial SOC of each cell to be up to $\pm 5\%$ away from $\bar{s} = 0.5$. For comparison, a centralized MPC (CMPC) and a cooperative distributed MPC (DMPC) [7] are also implemented. The simulation results are plotted in Figs. 2-5. In particular, Fig. 2 plots the SOC and control input for each cell for CMPC, Fig. 3 for AsynMPC, and Fig. 4 for DMPC. As can be seen, both AsynMPC and CMPC produce very comparable control performance. In both cases, the battery reach a balanced cell SOC at around 400 seconds. It can also be noted that the transient performance of the proposed AsynMPC is slightly worse compared to the CMPC. This is because only a few cells are optimized at each time instance, while for other cells, the previously optimized control commands are shifted for real-time usage. Such a decrease in control authority can be seen from Fig. 2(b) and Fig. 3(b), where AsynMPC commands much less balancing current compared to CMPC. DMPC, on the other hand, has a very comparable transient performance compared to CMPC as can be seen from Fig 4. However, it suffers from local minimum in this case and does not converge to the true optimum. Fig. 5 compares the standard deviation of cell SOC with respect to time for each controllers, where the degradation of AsynMPC during transient is clearly seen. However, such an acceptable degradation is rewarded by the large computation reduction as discussed next.

Overall, for AsynMPC, an average of less than 3 cells are optimized at each time step. As demonstrated in Table I, where computation time is measured in a standard PC with 16 GB Ram and an i-7 2.1 GHz processor, over 86% computation reduction can be achieved by the proposed AsynMPC approach. It is worth noting that, AsynMPC achieves very comparable computation reduction compared to DMPC, while significantly improving the control performance as discussed above.

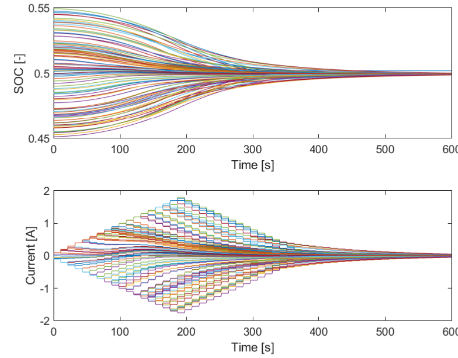

Fig. 3. Simulation results using the proposed AsynMPC. (a) SOC for each cell. (b) Balancing current for each cell.

TABLE I
COMPARISON OF COMPUTATION TIME

Controller	Solver Time [ms]	Relative Solver Time
CMPC	139.6	100%
AsynMPC	19.2	13.72%
DMPC	11.0	7.88%

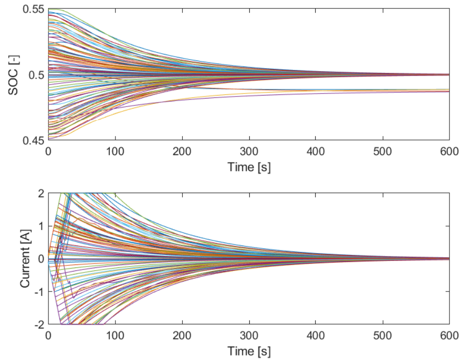

Fig. 4. Simulation results using cooperative distributed MPC. (a) SOC for each cell. (b) Balancing current for each cell.

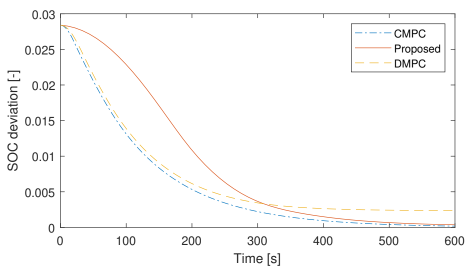

Fig. 5. Comparison of the standard deviations of cell SOC when balanced by different MPCs.

V. Conclusion

This paper proposes asynchronous model predictive control (MPC) for large interconnected systems subject to external disturbances, where the structure of the optimal control problem can be time-varying. In particular, the MPC is reconfigured at each time step to solve a relatively smaller optimization problem to reduce computation. Recursive feasibility and stability of the proposed approach are rigorously analyzed, hence supplementing existing work in this emerging area. In addition, a new reconfiguration policy that explicitly considers the prediction error introduced by external disturbance is also proposed. The advantage is clearly demonstrated through a numerical example on battery cell-to-cell balancing control with 100 cells connected in series. Specifically, the proposed asynchronous MPC can achieve over 86% computation reduction while keeping the closed-loop control performance comparable to centralized MPC. Future work will focus on observer-based control and its impacts on the reconfiguration policy and closed-loop stability.

References

[1] 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.

[2] 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.

[3] D. Bernardini and A. Bemporad, "Stabilizing model predictive control of stochastic constrained linear systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1468–1480, 2011.

[4] 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 Trans. Autom. Control*, vol. 56, no. 7, pp. 1524–1534, 2011.

[5] 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.

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

[7] 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 Eng.*, vol. 51, pp. 21–41, 2013.

[8] 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.

[9] 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.

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

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

[12] 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.

[13] 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.

[14] J. Chen, L. Zhang, and W. Gao, "Reconfigurable model predictive control for large scale distributed systems," *IEEE Systems Journal*, vol. 18, no. 2, pp. 965–976, June 2024.

[15] 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.

[16] D. Liu, B. Besselink, S. Baldi, W. Yu, and H. L. Trentelman, "An adaptive disturbance decoupling perspective to longitudinal platooning," *IEEE Control Systems Letters*, vol. 6, pp. 668–673, 2021.

[17] J. Chen, A. Behal, and C. Li, "Active battery cell balancing by real time model predictive control for extending electric vehicle driving range," *IEEE Trans. Auto. Sci. Eng.*, vol. 21, no. 3, pp. 4003–4015, July 2024.

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

[19] A. Ma, K. Liu, Q. Zhang, T. Liu, and Y. Xia, "Event-triggered distributed mpc with variable prediction horizon," *IEEE Transactions on Automatic Control*, vol. 66, no. 10, pp. 4873–4880, October 2021.

[20] Y. Jiang, J. Oravec, B. Houska, and M. Kvasnica, "Parallel MPC for linear systems with input constraints," *IEEE Transactions on Automatic Control*, vol. 66, no. 7, pp. 3401–3408, 2020.

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

[22] F. Dang, D. Chen, J. Chen, and Z. Li, "Event-triggered model predictive control with deep reinforcement learning," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 459–468, January 2024.

[23] 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.

[24] J. Chen, Z. Zhou, Z. Zhou, X. Wang, and B. Liaw, "Impact of battery cell imbalance on electric vehicle range," *Green Energy and Intelligent Transportation*, vol. 1, no. 3, pp. 1–8, December 2022.