Distributed and Localized Model Predictive Control—Part II: Theoretical Guarantees

Carmen Amo Alonso, Jing Shuang Li, Student Member, IEEE, Nikolai Matni, Senior Member, IEEE, and James Anderson, Senior Member, IEEE

Abstract—Engineered cyberphysical systems are growing increasingly large and complex. These systems require scalable controllers that robustly satisfy state and input constraints in the presence of additive noise: such controllers should also be accompanied by theoretical guarantees on feasibility and stability. In our companion paper, we introduced distributed and localized model predictive control (DLMPC) for large-scale linear systems; DLMPC is a scalable closed-loop MPC scheme in which subsystems need only exchange local information in order to synthesize and implement local controllers. In this article, we provide recursive feasibility and asymptotic stability guarantees for DLMPC. We leverage the system level synthesis framework to express the maximal positive robust invariant set for the closed-loop system and its corresponding Lyapunov function in terms of the closed-loop system responses. We use the invariant set as the terminal set for DLMPC, and show that this guarantees feasibility with minimal conservatism. We use the Lyapunov function as the terminal cost, and show that this guarantees stability. We provide fully distributed and localized algorithms to compute the terminal set offline, and also provide necessary additions to the online DLMPC algorithm to accommodate coupled terminal constraint and cost. In all algorithms, only local information exchange is necessary, and computational complexity is independent of the global system size: we demonstrate this analytically and experimentally. This is the first distributed MPC approach that provides minimally conservative yet fully distributed guarantees for recursive feasibility and asymptotic stability in the nominal and robust settings.

Index Terms—

I. INTRODUCTION

Ensuring recursive feasibility and asymptotic stability for MPC is a well-studied topic in the centralized setting [1], and sufficient conditions based on terminal sets and cost functions have been established [2]. Porting these ideas to distributed systems is a challenging task, both theoretically and computationally. High computational demand, limited and local communication, and coupling among subsystems prevent the use of techniques from the centralized setting. Thus, efforts have been made to develop theoretical guarantees for distributed MPC.

Prior Work: The majority of distributed MPC approaches rely on the use of distributed terminal costs and terminal sets to provide theoretical guarantees. In order to obtain structure in the terminal cost, standard methods rely on Lyapunov stability results, often combined with relaxation techniques, to make them amenable to distributed settings [3], [4], [5]. For terminal sets, proposed distributed methods are often limited by the coupling among subsystems: this often leads to small terminal sets that result in too conservative solutions (see, for example, [6] and [7]). In order to overcome these issues, several approaches have recently been proposed in this article, to synthesize structured terminal sets with adaptive properties, i.e., local terminal sets defined as the sublevel set of a structured Lyapunov function, which change at each iteration in order to avoid unnecessary conservatism [8], [9], [10], [11], [12], [13]. These approaches successfully design structured robust positive invariant sets and reduce conservatism; however, they require online updates of the terminal set at each MPC iteration, which increase the controller’s overall computational complexity and communication overhead. Moreover, the imposed structure unavoidably leads to a possibly small approximation of the maximal control invariant set (the least conservative option for a terminal set). To move away from structured sets and costs, a data-driven approach was recently developed in [14], where locally collected data are used to construct local control invariant sets and costs that provide guarantees. However, this approach is mainly limited to iterative control tasks, and conservatism of the terminal cost and set only reduces asymptotically as the system collects data.
Online computation and refinements of the terminal set are also key to this approach.

Given the state of the art, our goal is to design a distributed MPC approach with distributed and minimally conservative feasibility and stability guarantees. We seek a distributed MPC algorithm with: 1) a maximal positive invariant terminal set and 2) a fully distributed and scalable offline algorithm to compute this set. This will allow us to use the associated Lyapunov function of the terminal set as the terminal cost, without requiring explicit a priori structural assumptions. No method satisfying these requirements currently exists in literature.

Contributions: We provide recursive feasibility and asymptotic stability guarantees for the distributed localized MPC (DLMPC) method presented in our companion paper [15], for nominal and robust MPC (robust to additive polytopic or locally norm-bounded disturbances). This consists of two key contributions. Our first key contribution is to provide the first exact, fully distributed computation of the maximal positive invariant set, or terminal set. We show that it can be expressed in terms of the closed-loop system responses, as defined in the system level synthesis (SLS) framework [16]. Then, we show that when the closed-loop system is localized, the set is naturally structured without requiring additional assumptions; we leverage this to provide a distributed and localized offline algorithm for computation of the set. Our second key contribution is to provide the necessary additions to the original DLMPC algorithm to incorporate the coupled terminal constraint and terminal cost (i.e., global Lyapunov function [17]) associated with the maximal positive invariant set: this provides the algorithm with feasibility and stability guarantees, respectively. We provide additions to the algorithm to accommodate coupled terminal constraint and cost: this is done by using a nested alternating direction method of multipliers (ADMM)-based consensus algorithm. Overall, throughout all proposed algorithms, only local information exchanges within local neighborhoods are necessary, and computational complexity is independent of the global system size; each subcontroller first solves for its local portion of the terminal set, offline, then solves a local online MPC problem. We validate these results and further confirm the minimal conservatism introduced by this method through simulations.

Paper Structure: The rest of this article is organized as follows. In Section II, we present the problem formulation and briefly summarize essential concepts from our companion paper [15]. In Section III, we formulate the maximal robust positive invariant set and its associated Lyapunov function in closed-loop coordinates, and use this to provide recursive feasibility and stability guarantees for DLMPC; we also discuss convergence guarantees for DLMPC. In Section IV, we provide an offline algorithm to distributedly and locally compute the terminal set, and a modified online DLMPC algorithm that accommodates the terminal set and cost, including any local coupling. In Section V, we present a numerical study. Finally, Section VI concludes this article.

Notation: Lowercase and uppercase letters, such as $x$ and $A$, denote vectors and matrices, respectively, although lowercase letters might also be used for scalars or functions (the distinction will be apparent from the context). Bracketed indexes denote the time-step of the real system, i.e., the system input is $u(t)$ at time $t$, not to be confused with $x_t$, which denotes the predicted state $x$ at time $t$. Superscripted variables, e.g., $x^k$, correspond to the value of $x$ at the $k$th iteration of a given algorithm. The square bracket notation, i.e., $[x]$ denotes the components of $x$ corresponding to subsystem $i$. Calligraphic letters, such as $\mathcal{S}$, denote sets, and lowercase script letters, such as $\mathcal{C}$, denote a subset of $\mathbb{Z}^+$. Boldface lowercase and uppercase letters, such as $\mathbf{x}$ and $\mathbf{K}$, denote finite horizon signals and block lower triangular (causal) operators, respectively

$$
\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_T \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} K_0[0] & K_1[1] & K_1[0] \\ \vdots & \vdots & \vdots \\ K_T[T] & \ldots & K_T[1] & K_T[0] \end{bmatrix}
$$

where each $x_i$ is an $n$-dimensional vector, and each $K_i[j]$ is a matrix representing the value of $K$ at the $j$th time-step computed at time $i$. $\mathbf{K}(t, \mathcal{C})$ denotes the submatrix of $\mathbf{K}$ composed of the rows and columns specified by $t$ and $\mathcal{C}$, respectively. We denote the block columns of $\mathbf{K}$ by $\mathbf{K}\{1, \ldots, T\}$, i.e., $\mathbf{K}(1) := [K_0[0]^T \ldots K_T[T]^T]^T$, and we use $:=$ to indicate the range of columns, i.e., $\mathbf{K}\{2 : T\}$ contains the block columns from the second to the last. For compactness, we also define $Z_{AB} := \begin{bmatrix} I - Z \hat{A} & -Z \hat{B} \end{bmatrix}$, where $\hat{A} := \text{blkdiag}(A, \ldots, A)$ and $\hat{B} := \text{blkdiag}(B, \ldots, B, 0)$ for the dynamics matrices $A$ and $B$, and $Z$ is the block-downshift matrix.

II. Problem Formulation

We begin with a brief summary of the DLMPC formulation, approach, and algorithm (for details, refer to our companion paper [15]), then formally introduce the problem of providing theoretical guarantees—recursive feasibility and asymptotic stability—in the DLMPC scheme. We then introduce the open question that we resolve in this article.

Setup: Consider a discrete-time linear time invariant (LTI) system

$$
x(t + 1) = Ax(t) + Bu(t) + w(t) \tag{1}
$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^p$ is the control input, and $w(t) \in \mathcal{W} \subset \mathbb{R}^n$ is an exogenous disturbance at time $t$. System (1) can be interpreted as $N$ interconnected subsystems. Each subsystem is equipped with a subcontroller. We model the interconnection topology as an unweighted directed graph $\mathcal{G}_{(A, B)}(\mathcal{E}, \mathcal{V})$, where each subsystem $i$ is identified with a vertex $v_i \in \mathcal{V}$ and an edge $e_{ij} \in \mathcal{E}$ exists whenever $[A]_{ij} \neq 0$ or $[B]_{ij} \neq 0$.

We impose that the information exchange between subcontrollers—as defined by the graph $\mathcal{G}_{(A, B)}$—is confined to a subset of neighboring subcontrollers. In short, we restrict subcontrollers to exchange their state and control actions with neighbors at most $d$-hops away, as measured by the communication topology $\mathcal{G}_{(A, B)}$. We use the $d$-local communication constraints [16] to formalize this idea. Each subsystem $i$
1) receives information from its \(d\)-incoming set \(\text{in}_i(d)\):
\[
\text{in}_i(d) := \{v_j \mid \text{dist}(v_j \rightarrow v_i) \leq d \in \mathbb{N}\};
\]
2) sends information to its \(d\)-outgoing set \(\text{out}_i(d)\):
\[
\text{out}_i(d) := \{v_j \mid \text{dist}(v_i \rightarrow v_j) \leq d \in \mathbb{N}\}.
\]

**Remark 1:** Notice that when \(\text{in}_i(d) = \text{out}_i(d)\) for all \(i\), the communication network is undirected.

We use a model predictive controller to determine the control input; at time step \(\tau\), the controller solves
\[
\min_{x_\tau, u_\tau, \gamma_\tau} \sum_{t=0}^{T-1} f_1(x_t, u_t) + f_T(x_T)
\]
subject to
\[
x_0 = x(\tau), \quad x_{t+1} = Ax_t + Bu_t + w_t
\]
and
\[
x_T \in X_T, \quad x_t \in X_t, \quad u_t \in U_t \forall w_t \in W_t \quad (2)
\]
To provide tractability, \(f_1(\cdot, \cdot)\) and \(f_T(\cdot)\) are assumed to be closed, proper, and convex, and \(\gamma_t(\cdot)\) is a measurable function of its arguments. The sets \(X_t\) and \(U_t\) are assumed to be closed and convex sets containing the origin for all \(t\). For simplicity, we will consider constant state and input constraint sets \(X\) and \(U\).

Note that (2) does not include the \(d\)-local communication constraints. It is not possible to introduce such constraints in a convex manner in the classical MPC formulation (2). However, the DLMPC formulation allows to incorporate locality constraints into the MPC formulation in a straightforward manner with the only requirement that the MPC formulation be compatible with the locality constraints. (see our companion work [15] for details). Hence, we assume that if two subsystems are coupled through either the constraints or cost, then the two subsystems must be in the \(d\)-incoming and \(d\)-outgoing set from one another:

**Assumption 1:** Given an MPC problem (2) over system (1), the objective function \(f\) is such that \(f(x, u) = \sum f_i([x]_{\text{in}_i(d)}, [u]_{\text{in}_i(d)})\) for \(f_i\) local functions; and the constraint sets are such that \(x \in X = X^1 \cap \ldots \cap X^N\), where \(x \in X\) if and only if \([x]_{\text{in}_i(d)} \in X^i\) for all \(i\), and \(x \in X^i\) and \(u \in U^i\).

**Approach:** In [15], we use the SLS framework to reformulate the MPC problem (2) into the DLMPC problem. This allows for distributed and localized synthesis and implementation, i.e., each subsystem requires only local information to synthesize its local subcontroller and determine the local control action. This is made possible by imposing appropriate \(d\)-local structural constraints \(L_d\) on the close-loop system responses of the system \(\Phi_x\) and \(\Phi_u\), which become the decision variables of the MPC problem.

The DLMPC subroutine over time horizon \(T\) at time \(\tau\) is as follows:
\[
\min_{\Phi} f(\Phi_1\{1\}x_0) + f_T(\Phi_1\{1\}x_0)
\]
subject to
\[
Z_{AB} \Phi = I, \quad x_0 = x(\tau), \quad \Phi \in L_d, \quad \Phi w \in \mathcal{P}, \quad \Phi x, \Phi w \in X_T \forall w \in W \quad (3)
\]
where \(\Phi := [\Phi_x^\top, \Phi_u^\top]^\top\). \(f_1(\cdot, \cdot)\) and \(f_T(\cdot)\) are closed, proper, and convex cost functions, and \(\mathcal{P}\) is defined so that \(\Phi w \in \mathcal{P}\) if and only if \(x \in X\) and \(u \in U\). By convention, we define the disturbance to contain the initial condition, i.e., \(w = [x_0^T \quad w_0^T \ldots w_{T-1}^T] = [x_0^T \delta^T]\) and \(W\) is defined over \(\delta\) so that it does not restrict \(x_0\). In the nominal case, i.e., \(w_t = 0 \forall t, P\) and \(X_T\) are closed and convex sets containing the origin. When noise is present, we restrict ourselves to polytopic sets only: \(\mathcal{P} := \{x^T u^T \mid : H[x^T u^T]^T \leq h\}\), and consider different options for set \(W\):
"
III. FEASIBILITY AND STABILITY GUARANTEES

Theoretical guarantees for the DLMPC problem (3) are now derived. First, we describe a maximal positive invariant set using an SLS-style parametrization; recursive feasibility for DLMPC is guaranteed by using this set as the terminal set. We also use this set to construct a terminal cost to guarantee asymptotic stability for the nominal setting and input-to-state stability (ISS) for the robust setting. Convergence results from the ADMM literature are used to establish convergence guarantees.

A. Feasibility Guarantees

Recursive feasibility guarantees for the DLMPC problem (3) are given by the following lemma:

**Lemma 1:** Let the terminal set $X_T$ for the DLMPC problem (3) be of the form

$$X_T := \{x_0 \in \mathbb{R}^n : \Phi x_0^T \delta^T \in \mathcal{P} \forall \delta \in \mathcal{W}\}$$

for some $\Phi$ satisfying $Z_{AB} \Phi = I$. Then recursive feasibility is guaranteed for the DLMPC problem (3). Moreover, $X_T$ is the maximal robust positive invariant set for the closed-loop system described by $\Phi$.

**Proof:**

First, we show that $X_T$ is the maximal robust positive invariant set. By [1, Algorithm 10.4], the set

$$S := \bigcap_{k=0}^{\infty} S_k, \text{ with } S_0 = \mathcal{X}$$

$$S_k = \{x \in \mathbb{R}^n : Ax + Bu + w \in S_{k-1} \forall w \in \mathcal{W}\}$$

is the maximal robust positive invariant set for the closed-loop system (1) with some fixed input $u \in U$. We show by induction that $S_k$ can also be written as

$$S_k = \{x_0 \in \mathbb{R}^n : x \in \mathcal{X} \forall w_{0:k-1} \in \mathcal{W}\}$$

for some sequence $w_{0:k-1} \in U$. The base case at $k = 1$ is trivially true, since $S_0 = \mathcal{X}$. For the inductive step, assume that (6) holds at $k$. Then, at $k = 1$,

$$S_{k+1} = \{x \in \mathbb{R}^n : Ax + Bu + w \in S_k \forall w \in \mathcal{W}\}$$

$$= \left\{ x_0 \in \mathbb{R}^n : x := Ax_0 + Bu + w \text{ s.t.} A^{k+1}x + \sum_{j=0}^{k} A^j(Bu_j + w) \in \mathcal{X} \forall w \in \mathcal{W}\right\}$$

for some sequence of inputs $w_{0:k} \in U$. This implies that $S$ can be written as

$$S = \{x_0 \in \mathbb{R}^n : x \in \mathcal{X} \forall w \in \mathcal{W}\}$$

for some $\Phi$ satisfying $Z_{AB} \Phi = I$. This constraint automatically enforces that the resulting closed-loop is feasible, i.e., $u = \Phi u \left[x_0^T \delta^T\right]$ exists; furthermore, all inputs $u$ can be captured by this closed-loop parametrization since it parametrizes a linear time-varying controller over a finite time horizon [16]. Hence, $X_T$ is the maximal robust positive invariant set for the closed-loop system, as defined by $\Phi$.

Next, we show that imposing $X_T$ as the terminal set of the DLMPC problem (3) guarantees recursive feasibility. Notice that by definition, $X_T$ is not only a robust positive invariant set but also a robust control invariant set. We can directly apply the proofs from [1, Th. 12.1] to the robust setting: recursive feasibility is guaranteed if the terminal set is control invariant, which $X_T$ is.

**Remark 2:** Recursive feasibility is guaranteed by a control invariant $X_T$. In the ideal case, we want $X_T$ to be the maximal robust control invariant set, in order to minimize conservatism introduced by $X_T$ in (3). However, this set generally lacks sparsity, violates Assumption 1, and is not amenable for inclusion in our distributed and local algorithm. For this reason, we use the maximal robust positive invariant set in Lemma 1. Here, the choice of $\Phi$ is critical in determining how much conservatism $X_T$ will introduce. As suggested in [1, Sec. 12], we choose $\Phi$ corresponding to the unconstrained closed-loop system. In the interests of distributed synthesis and implementation, we additionally enforce $\Phi$ to have localized structure. We discuss how this $\Phi$ is computed in Section IV, and demonstrate that the resulting terminal set $X_T$ introduces no conservatism in Section V.

By Assumption 1, constraint sets are localized, i.e., $X = X^1 \cap \ldots \cap X^N$, where $x \in X$ if and only if $[x]_{\mathcal{IN}(d)} \in X^i$ for all $i$ (and idem for $U$). Moreover, we can use the constraint $L_d$ to enforce that the system response $\Phi$ is localized. This implies that the set $X_T$ is also localized

$$X = X^1_T \cap \ldots \cap X^N_T, \text{ where}$$

$$X^i_T = \{[x]_{\mathcal{IN}(d)} \in \mathbb{R}^{[n]^i} : [\Phi u]_{\mathcal{IN}(d)} \left[x_0^T \delta^T\right]_{\mathcal{IN}(d)} \in \mathcal{P}^i \forall [\delta]_{\mathcal{IN}(d)} \in \mathcal{W}^{\mathcal{IN}(d)}\}$$

and $x \in X_T$ if and only if $[x]_{\mathcal{IN}(d)} \in X^i_T$ for all $i$. We will show in Section IV that this allows for a localized and distributed computation of the terminal set $X_T$.

B. Stability Guarantees

Stability guarantees for the DLMPC problem (3) are given by the following lemma.

**Lemma 2:** Consider system (1) subject to the MPC law (3), where:

1) the cost $f$ is continuous and positive definite;
2) the set $\mathcal{P}$ contains the origin and is closed;
3) $X_T$ is defined by (5);
4) $f_T(x) = \inf \{\eta \geq 0 : x \in \eta X_T\}$.

Then, in the nominal setting, the origin is asymptotically stable with domain of attraction $\mathcal{X}$. In the robust setting, $X_T$ is input-to-state stable with domain of attraction $\mathcal{X}$.
Proof: It suffices to show that these conditions immediately imply satisfaction of the necessary assumptions in [20], together with the additional sufficient conditions of [21, Th. 4.2]. These results state that if

i) \( f \) and \( f_T \) are continuous and positive definite; ii) \( \mathcal{X}, \mathcal{U}, \mathcal{X}_T \) contain the origin and are closed; iii) \( \mathcal{X}_T \) is control invariant; iv) \( \min_{\mathcal{X}_T} f_T(x) \leq 0 \forall x \in \mathcal{X}_T; \)

then the desired stability guarantees hold.

Condition 1 implies satisfaction of the part of (i) that concerns \( f \). Condition 2) implies satisfaction of (ii). If \( \mathcal{P} \) contains the origin and is closed, by definition this implies \( \mathcal{X} \) and \( \mathcal{U} \) also contain the origin and are closed. Also, since \( \mathcal{X}_T \) is defined in terms of \( \mathcal{P} \) in (5), \( \mathcal{X}_T \) also contains the origin and is closed. Condition 3) implies satisfaction of (iii) by virtue of Lemma 1. Condition 4) implies that \( f_T \) is a Lyapunov function on \( \{ x \in \mathbb{R}^n : 1 \leq f_T(x) \} \supseteq \mathcal{X}_T \) since it is the Minkowski functional of the terminal set \( \mathcal{X}_T \) (see [17, Th. 3.3.]). Therefore, the condition stated in (iv) is automatically satisfied for all \( x \in \mathcal{X}_T \). Moreover, \( f_T \) is necessarily positive definite, so the part of (i) that concerns \( f_T \) is satisfied as well.

Therefore, by virtue of the results in [20] and [21], we guarantee asymptotic stability of the origin in the nominal setting and ISS of \( \mathcal{X}_T \) in the robust setting, both with domain of attraction \( \mathcal{X} \).

For any cost \( f \) and constraint \( \mathcal{P} \) that satisfy conditions 1) and 2) of Lemma 2, we can choose an appropriate terminal set \( \mathcal{X}_T \) and terminal cost \( f_T \) as per 3) and 4) to satisfy the lemma. This guarantees stability for the DLMPC problem (3). However, as stated, \( f_T \) does not satisfy Assumption 1, and therefore it is not localized. In particular, \( f_T \) can be written as

\[
 f_T(x) = \inf_{\eta \geq 0} \left\{ \eta \geq 0 : [x]_{\text{in}\,(\mathcal{P})} \in \eta \mathcal{X}_T^\perp \forall \epsilon \right\} \tag{7}
 \]

which cannot be written as a sum of local functions. Nonetheless, this scalar objective function admits a distributed and localized implementation: we can add it in the DLMPC algorithm using the ADMM-based consensus technique described in Section IV.

C. Convergence Guarantees

Algorithm 1 relies on ADMM. We can guarantee convergence of the overall algorithm by leveraging the ADMM convergence result from [18].

Lemma 3:

In Algorithm 1, the residue, objective function, and dual variable converge as \( k \to \infty \), i.e.,

\[
\tilde{H}(k) - \tilde{\Phi}(k) \to 0, f(\tilde{\Phi}(k), x_0) \to f(\tilde{\Phi}^*, x_0), \Psi(k) \to \Psi^*
\]

where \( * \) indicates optimal value.

Proof: Algorithm 1 is the result of applying ADMM to the DLMPC problem (3), then exploiting the separability and structure of resulting subproblems to achieve distributed and localized implementation, as presented in [15]. Thus, to prove convergence, we only need to show that the underlying ADMM algorithm converges. By the ADMM convergence result in [18], the desired convergence of the residue, objective function, and dual variable are guaranteed if, for this algorithm,

1) the extended-real-value functional is closed, proper, and convex; and

2) the unaugmented Lagrangian has a saddle point.

We first show 1). The extended-real-value functional \( h(\tilde{\Phi}) \) is defined for this algorithm as

\[
h(\tilde{\Phi}) = \begin{cases} f(M_1 \tilde{\Phi}(k), x_0), & \text{if } Z_{AB} M_2 \tilde{H} = I \text{ and } \tilde{\Phi} \in \mathcal{L}_d, \tilde{\Phi}(k) \in \mathcal{P} \\ \infty, & \text{otherwise}. \end{cases}
\]

where \( \tilde{H}^* \) is the left inverse of \( \tilde{H} \) from (4).

By assumption, \( f(M_1 \tilde{\Phi}(k), x_0) \) is closed, proper, and convex, and \( \mathcal{P} \) is a closed and convex set. The remaining constraints \( Z_{AB} \tilde{\Phi} = I \) and \( \tilde{\Phi} \in \mathcal{L}_d \) are also closed and convex. Hence, \( h(\tilde{\Phi}) \) is closed, proper, and convex.

We now show 2). This condition is equivalent to showing that strong duality holds [22]. Since problem (3) is assumed to have a feasible solution in the relative interior of \( \mathcal{P} \) by means of Lemma 1 (given that the first iteration is feasible), Slater’s condition is automatically satisfied, and therefore the unaugmented Lagrangian of the problem has a saddle point.

Both conditions of the ADMM convergence result from [18] are satisfied; Algorithm 1 converges in residue, objective function, and dual variable.

Remark 3: It is important to note that the communication network might be a directed graph: depending on the elements in the incoming as outgoing sets. The proof outlined here holds as well for the case where the communication graph is directed. For additional details on ADMM convergence over directed networks, see [23].

IV. Feasible and Stable DLMPC

We incorporate theoretical results from Section III into the DLMPC computation. We provide an algorithm to compute the terminal set \( \mathcal{X}_T \). We also provide a distributed and localized computation for the terminal cost function \( f_T \). The terminal set and cost generally introduce local coupling among subsystems; this minimizes conservatism but requires an extension to Algorithm 1 to accommodate coupling. All algorithms provided are distributed and localized, with computational complexity that is independent of the global system size.

A. Offline Synthesis of the Terminal Set \( \mathcal{X}_T \)

Our first result is to provide an offline algorithm to compute terminal set \( \mathcal{X}_T \) from (5) in a distributed and localized manner. As discussed in Section III, we compute the maximal robust positive invariant set for the unconstrained localized closed-loop system. We use SLS-based techniques to obtain a localized closed-loop map \( \tilde{\Phi} \).
Our algorithm is based on [1, Algorithm 10.4]. The advantage of implementing this algorithm in terms of the localized closed-loop map $\Phi$ is two-fold: 1) we can work with locally bounded and polytopic disturbance sets $W$ by leveraging [15, Lemmas 1 and 2], and 2) the resulting robust invariant set is automatically localized given the localized structure of the closed-loop map.

We start by finding a localized closed-loop map $\Phi$ for system (1). To do this, we need to solve
\[
\min_{\Phi} f(\Phi) \quad \text{s.t. } Z_{AB} \Phi = I, \Phi \in L_d. \tag{8}
\]
This is an SLS problem with a separable structure for most standard cost functions $f$ [16]. The separable structure admits localized and distributed computation. Even when separability is not apparent, a relaxation can often be found that allows for distributed computation (see, for example [24]). The infinite-horizon solution for quadratic cost is presented in [25]. For other costs, computing an infinite horizon solution to (8) remains an open question; in these cases, we can use finite impulse response SLS with sufficiently long time horizon.

Once a localized closed-map $\Phi$ has been found, we can compute the associated maximal positive invariant set. In the robust case, we leverage results from Lemmas 1 and 2 in our companion paper [15], which use duality arguments to tackle specific formulations of $W$. We denote $\sigma$ as the upper bound of $\|x\|_s$ for all $i$, where $\|\cdot\|_s$ is the dual norm of $\|\cdot\|_u$. Also, $e_j$ is the $j$th vector in the standard basis.

**Nominal (i.e., no disturbance)**

$$S := \{ x_0 \in \mathbb{R}^n : \Phi \{ 1 \} x_0 \in \mathcal{P} \}.$$

**Locally bounded disturbance**

$$S := \{ x_0 \in \mathbb{R}^n : [H_j \Phi \{ 1 \} x_0]_i \leq \sigma \|e_j^T [H_j \Phi \{ 2 : T \}]_i\|_s \leq [\hat{h}]_i, \forall i \}.$$

**Polytopic disturbance**

$$S := \{ x_0 \in \mathbb{R}^n : H \Phi \{ 1 \} x_0 + \Xi g \leq \hat{h}, \forall i \}.$$

As per Lemma 1, we can write $S$ by iteratively computing $S_{k+1}$ from $S_k$.

Assume $S_k$ can be written as

$$S_k := \{ x_0 \in \mathbb{R}^n : \hat{H} x \leq \hat{h} \}.$$

Then, we can write $S_{k+1}$ as

$$S_{k+1} := \{ x_0 \in \mathbb{R}^n : \Phi_{x, 1} [1] x_0 \in \mathcal{F}(S_k) \}.$$

**Algorithm 2: Subsystem $i$ Terminal Set Computation.**

| input: $\Phi_{x, i} [1] \in \mathcal{F}(d), \Phi_{u, 0} [0] \in \mathcal{F}(d), \mathcal{U}^\text{in}(d)$ for polytopic noise, also $[G]_\text{in}(d), [g]_\text{in}(d)$ | output: $x_T^i, \mathcal{F}(S_k)$ |
|---|---|
| 1: $S_0^i \leftarrow x_T^i, k \leftarrow -1$ | 1: $S_0^i \leftarrow x_T^i, k \leftarrow -1$ |
| 2: repeat | 2: repeat |
| 3: $k \leftarrow k + 1$ | 3: $k \leftarrow k + 1$ |
| 4: Share $S_{k-1}^i$ with out$_i(d)$. | 4: Share $S_{k-1}^i$ with out$_i(d)$. |
| Receive $S_{k}^j$ from all $j \in \text{in}_i(d)$. | Receive $S_{k}^j$ from all $j \in \text{in}_i(d)$. |
| 5: Compute $S_{k+1}^i$ via (10). | 5: Compute $S_{k+1}^i$ via (10). |
| 6. $S_{k+1}^i \leftarrow S_{k}^i \cap S_{k+1}^i$ | 6. $S_{k+1}^i \leftarrow S_{k}^i \cap S_{k+1}^i$ |
| 7: until: $S_{k+1}^i = S_k^i$ for all $i$ | 7: until: $S_{k+1}^i = S_k^i$ for all $i$ |

where $\mathcal{F}(S_k) := S_k$ in the nominal case. In the case of locally bounded disturbance

$$\mathcal{F}(S_k) := \{ x \in \mathbb{R}^n : \hat{H} x + \Xi g \leq \hat{h} \}.$$

and for polytopic disturbance

$$\mathcal{F}(S_k) := \{ x \in \mathbb{R}^n : \hat{H} x + \Xi g \leq \hat{h}, \forall i \}.$$

These formulations use the simplifying fact that $\Phi_{x, 0} [0] = I$ for all feasible closed-loop dynamics. Also, notice that by using $S_k$ to calculate $S_{k+1}$, the only elements of $\Phi$ that we require are $\Phi_{x, 1} [1]$ and $\Phi_{u, 0} [0]$.

The conditions stated in (9) are row- and column-wise separable in $\Phi$ and $\Xi$; this allows us to calculate $S_k$ in a distributed way. Furthermore, $\mathcal{F}(S_k)$ and $\mathcal{U}$ are localizable by Assumption 1, in both nominal and robust settings. Since $\Phi$ is also localized, we can exploit the structure of $\Phi$ to and rewrite $S_{k+1}$ as the intersection of local sets, i.e., $S_{k+1} = S_{k+1}^1 \cap \cdots \cap S_{k+1}^N$, where

$$\Xi(j, :) = \min_{\Xi_{j, g} \geq 0} \Xi_{j, g} \text{s.t. } \hat{H} (j, :) \Xi_{j, g} = \Xi_{j, G} \forall j.$$

These formulations use the simplifying fact that $\Phi_{x, 0} [0] = I$ for all feasible closed-loop dynamics. Also, notice that by using $S_k$ to calculate $S_{k+1}$, the only elements of $\Phi$ that we require are $\Phi_{x, 1} [1]$ and $\Phi_{u, 0} [0]$.

The conditions stated in (9) are row- and column-wise separable in $\Phi$ and $\Xi$; this allows us to calculate $S_k$ in a distributed way. Furthermore, $\mathcal{F}(S_k)$ and $\mathcal{U}$ are localizable by Assumption 1, in both nominal and robust settings. Since $\Phi$ is also localized, we can exploit the structure of $\Phi$ to and rewrite $S_{k+1}$ as the intersection of local sets, i.e., $S_{k+1} = S_{k+1}^1 \cap \cdots \cap S_{k+1}^N$, where

$$S_{k+1} = \{ x_0 \in \mathbb{R}^n : \hat{H} x \leq \hat{h}, \forall i \}.$$

As per Lemma 1, we can write $S$ by iteratively computing $S_{k+1}$ from $S_k$.

Assume $S_k$ can be written as

$$S_k := \{ x_0 \in \mathbb{R}^n : \hat{H} x \leq \hat{h} \}.$$

Then, we can write $S_{k+1}$ as

$$S_{k+1} = \{ x_0 \in \mathbb{R}^n : \Phi_{x, 1} [1] x_0 \in \mathcal{F}(S_k) \}.$$

2For simplicity of presentation, we write out $S$ only for finite-horizon $\Phi$; the proposed algorithm to synthesize $S$ works for infinite-horizon $\Phi$ as well.

3For formal definitions of row and columnwise separability see [15].
set will be $2$-localized since—in the presence of coupling—Algorithm 2 requires communication with not only local patch neighbors but also neighbors of those neighbors. Convergence is guaranteed for system (1) if it is stable when $u = 0$, $w = 0$, and when the constraint and disturbance sets $X$, $\mathcal{U}$, $\mathcal{W}$ are bounded and contain the origin, as per [26].

Computational Complexity of the Algorithm: In Algorithm 2, step 4 refers to communication exchanges, while steps 5, 6, and 7 tackle computational steps. In terms of the computational complexity of Algorithm 2, the order of magnitude of the computations at each local subsystem $i$ is $O(m_i d_i)$, where $m_i$ is the number of constraints on subsystem $i$. Notice that in the case of a coupled scenario, the number of constraints at subsystem $i$, $m_i$, is in general larger than in the uncoupled case, since additional constraints couple neighboring subsystems. Notice also that the communication exchanges occur only within each $d$-local neighborhood, and therefore the computation of the maximal robust positive invariant set can be computed in a purely distributed and localized manner. Hence, its scalability is independent from the size of the whole system. It is also worth noting that this computation occurs offline, where both communication and computational complexity are not so critical.

B. Online Synthesis of DLMPC

DLMPC Algorithm 1 was developed under the assumption that no coupling is introduced through cost or constraints. We now extend this algorithm to accommodate local coupling, which is allowed as per Assumption 1. This allows us to incorporate the terminal set and cost introduced in the previous sections, both of which generally induce local coupling. We will use notation that assumes all constraints and costs (including the terminal set) are $d$-localized. In the case that the terminal set is $2$-localized, subsystems will need to exchange information with neighbors up to $2$-hops away; simply replace \{out$_i(d)$, in$_i(d)$\} with \{out$_i(2d)$, in$_i(2d)$\} wherever they appear in Algorithms 1, 2, and 3.

Appending the terminal set (5) and terminal cost (7) to the DLMPC problem (4), gives

$$
\min_{\Phi, \tilde{\Phi}, \eta} \quad f(M_1 \tilde{\Phi}\{1\}x_0) + \eta \\
\text{s.t.} \quad Z_{AB}M_2 \tilde{\Psi} = I, x_0 = x(\tau), \tilde{\Phi}, \tilde{\Psi} \in \mathcal{L}_d \\
\tilde{\Phi} \in \tilde{\mathcal{P}}, \tilde{H} \tilde{\Phi} = \tilde{\Psi} \\
M_1 \tilde{\Phi}_T\{1\}x_0 \in \eta \mathcal{X}_T, 0 \leq \eta \leq 1
$$

(11)

where $\tilde{\Phi}_T$ represents block rows of $\tilde{\Phi}$ that correspond to time horizon $T$, e.g., in the nominal setting, the last block row of $\tilde{\Phi}_x$, and the set $\tilde{\mathcal{P}}$ is defined so that translates the constraint on $\tilde{\Phi}_x \in \mathcal{P}$ into a constraint directly on $\tilde{\Phi}$.

**Remark 4:** In the robust setting, we incorporate the terminal constraint into $\tilde{\mathcal{P}}$ to ensure robust satisfaction of the terminal constraint. However, $\mathcal{X}_T$ still appears as nominal constraint in order to integrate the definition of the terminal cost (7) into the DLMPC formulation (3).

In the original formulation (4), we assume no coupling; as a result, all expressions involving $\tilde{\Phi}$ are row-separable, and all expressions involving $\tilde{\Psi}$ are column-separable. If we allow local coupling as per Assumption 1, we lose row-separability; row-separability is also lost in the new formulation (11) due to local coupling in the terminal set. This is because without coupling, $[x_i]$ and $[u_i]$ (which correspond to a fixed set of rows in $\tilde{\Phi}$), can only appear in $f^j$ and $\mathcal{P}^i$; thus, each row of $\tilde{\Phi}$ (and $\tilde{\Psi}$) is solved by exactly one subsystem in step 3 of Algorithm 1. With coupling, $[x_i]$ and $[u_i]$ can appear in $f^j$ and $\mathcal{P}^j$ for any $j \in \text{in}_i(d)$; now, each row of $\tilde{\Phi}$ is solved for by multiple local subsystems at once, and row-separability is lost. This only affects step 3 of Algorithm 1 (i.e., the rowwise problem); other steps remain unchanged. We write out the rowwise problem corresponding to (11)

$$
\min_{\Phi} \quad f(M_1 \Phi\{1\}x_0) + \eta + g(\Phi, \tilde{\Phi}, \tilde{\Psi}, \Lambda) \\
\text{s.t.} \quad \tilde{\Phi} \in \tilde{\mathcal{P}} \cap \mathcal{L}_d, M_1 \tilde{\Phi}_T\{1\}x_0 \in \eta \mathcal{X}_T \\
0 \leq \eta \leq 1, x_0 = x(\tau)
$$

(12)

where $g(\Phi, \tilde{\Phi}, \tilde{\Psi}, \Lambda) = \frac{\eta}{2}\|\tilde{H} \tilde{\Phi} + \Lambda\|^2_F$, and $\Lambda$ is the additional variable introduced by ADMM. For additional details, see [15, eq. (10)].

Note that if $\mathcal{P}$ induces coupled linear inequality constraints, row-separability is maintained. Though $[x_i]$ and $[u_i]$ appear in multiple $\mathcal{P}_{j}$, this corresponds to distinct rows of $\Omega$, each solved by one subsystem, as opposed to one row of $\Phi$ that is solved for by multiple subsystems. A similar idea applies if coupling is induced by some quadratic cost, i.e., $f(\Phi x_0) = \|C \Phi x_0\|_F^2$, for some matrix $C$ which induces coupling. In this case, we can modify $\tilde{H}$ such that we enforce $\Phi = C \tilde{\Psi}$, and rewrite the cost as $\|\Phi x_0\|_2^2$; now, each row of $\Phi$ is solved by exactly one subsystem, and row-separability is maintained. However, this technique does not apply to coupling induced by the terminal cost: it is generally true that each row of $\Phi$ will be need to be solved for by several subsystems. We introduce a new vector variable $X := M_1 \Phi\{1\}x_0$.

This variable facilitates consensus between subsystems who share the same row(s) of $\Phi$. Each subsystem solves for components $\{X|\text{in}_i(d)\}$, and comes to a consensus with its neighboring subsystems on the value of these components. In the interest of efficiency, we directly enforce consensus on elements of $X$ instead of enforcing consensus on rows of $\Phi$. We introduce a similar variable for terminal cost $\eta_i$; we define vector $\eta := [\eta_1, \ldots, \eta_N]$. Subsystem $i$ solves for $\eta_i$, which is its own copy of $\eta$. It then comes to a consensus on this value with its neighbors, i.e., $\eta_j$ has the same value $\forall j \in \text{in}_i(d)$. Assuming that $\mathcal{G}_{(A,B)}$ is connected, this guarantees that $\eta_i$ has the same value $\forall i \in \{1, \ldots, N\}$, i.e., all subsystems agree on the value of $\eta$. We combine these two consensus-facilitating variables into

\footnote{We would also need to use $[\Psi_{u,0}]\{0\}$, for the algorithm output instead of $[\Phi_{u,0}]\{0\}$.}
the augmented vector variable
\[ \tilde{X} := [X^\top \eta^\top]^\top. \]

With this setup, we follow a variable duplication strategy and apply ADMM for consensus [27]. In particular, we duplicate variables so each subsystem has its own copy of the components of \( \tilde{X} \). Problem (12) becomes
\[
\min_{\tilde{\Phi}, \tilde{X}, \tilde{Y}} \tilde{f}(\tilde{X}) + g(\tilde{\Phi}, \tilde{\Psi}, \Lambda^k) \\
\text{s.t.} \quad \tilde{\Phi}, \tilde{X} \in Q, \quad \tilde{X} \in \tilde{X}_T, \quad \tilde{\Phi} \in \mathcal{L}_d, \quad x_0 = x(\tau) \\
[\tilde{X}^j]_i = [M_i \tilde{\Phi}]_i, [x_0]^j_i, [\tilde{X}^j]_j = [\tilde{Y}^j]_j \quad \forall j \in \mathcal{i}_d(d) \\
\forall i, \,
\] (13)

where we define \( \tilde{f}, \tilde{X}_T, \) and \( Q \) as follows:
1) \( \tilde{f}(\tilde{X}) := f(X) + \frac{\mu}{2} \sum_{i=1}^N \eta_i; \)
2) \( \tilde{X} \in \tilde{X}_T \) if and only if \( M_i \tilde{\Phi}_T \{1\} x_0 \in \eta_i \mathcal{X}_T \forall i; \)
3) \( \tilde{\Phi}, \tilde{X} \in Q \) if and only if \( \eta \leq 1 \) and \( \tilde{\Phi} \) induces linear inequalities or \( \tilde{X} \in \tilde{P} \) otherwise.\(^5\)

The structure of problem (13) allows us to solve it in a distributed and localized manner via ADMM-based consensus. In particular, each subsystem \( i \) solves
\[
\begin{align*}
\left[ \tilde{\Phi}^i_{k+1,n+1} \right]_{\mathcal{i}_d(d)} & = \arg \min_{\tilde{\Phi}^i_{k,n+1}} \tilde{g}^i_k(\tilde{\Phi}^i, \tilde{\Phi}^n, \tilde{Z}^n) \\
\left[ \tilde{X}^n_{\mathcal{i}_d(d)} \right]_{\mathcal{i}_d(d)} & = \arg \min_{\tilde{X}^n} h^i(\tilde{X}^n, \tilde{\Phi}_r, \tilde{\Psi}_r, \Lambda^i) \\
\end{align*}
\] (14a)
\[
\text{s.t.} \quad [\tilde{\Phi}]^i_\mathcal{i}_r \in \mathcal{Q}^i, [\tilde{X}^i_\mathcal{i}_r] \in \tilde{X}^i_\mathcal{i}_r \cap \mathcal{Q}^i, [\tilde{X}^i_\mathcal{i}_r]_i = [M_i \tilde{\Phi}]_i, [x_0]^i_i \\
\] (14b)
\[
\left[ \tilde{Y}^n_{\mathcal{i}_d(d)} \right]_{\mathcal{i}_d(d)} = \frac{h^i(\tilde{X}^n_{\mathcal{i}_d(d)}, \tilde{\Phi}_r, \tilde{\Psi}_r, \Lambda^i)}{[\mathcal{i}_d(d)]} \\
\] (14c)

where to simplify notation, we define
\[
\begin{align*}
\tilde{g}^i_k(\tilde{\Phi}^i, \tilde{\Phi}^n, \tilde{Z}^n) & := \tilde{f}^i([X^i_{\mathcal{i}_d(d)}]_d) + g([\tilde{\Phi}]^i_\mathcal{i}_r, [\tilde{\Psi}]^i_\mathcal{i}_r, [\Lambda]^i_\mathcal{i}_r) \\
h^i(\tilde{X}^n, \tilde{\Phi}_r, \tilde{\Psi}_r, \Lambda^i) & := \sum_{j \in \mathcal{i}_d(d)} \|X^j_j - Y^j_j\|^2 + \|Z^j_j\|^2. \\
\end{align*}
\]
Consensus iterations are denoted by \( n \), outer-loop (i.e., Algorithm 1) iterations are denoted by \( k \), and \( \mu \) is the ADMM consensus parameter. Intuitively, \( \tilde{g}^i_k \) represents the original objective from (12), and \( h^i \) represents the consensus objective.

Algorithm 3: Subsystem \( i \) Implementation of Step 3 in Algorithm 1 When Subject to Localized Coupling.

**input:** tolerance parameters \( \epsilon_x, \epsilon_z, \mu > 0 \).

1: \( n \leftarrow 0 \);
2: Solve optimization problem (14a);
3: Share \( [X]^n_{i} + \text{with out}_i(d) \). Receive the corresponding \( [X]^n_{j} + \text{from } j \in \mathcal{i}_d(d) \).
4: Perform update (14b);
5: Share \( [Y]^n_{i} + \text{with out}_i(d) \). Receive the corresponding \( [Y]^n_{j} + \text{from } j \in \mathcal{i}_d(d) \).
6: Perform update (14c);
7: if \( \|X^n_{i} - Z^n_{i} + 1\|_F < \epsilon_x \) and \( \|Z^n_{i} - Z^n_{i} + 1\|_F < \epsilon_z \) :

Go to step 4 in Algorithm 1.

**else:** Set \( n \leftarrow n + 1 \) and return to step 2.

compared to the uncoupled scenario (i.e., Algorithm 1); however, the scalability properties from the uncoupled scenario still apply. Complexity in the current algorithm is determined by steps 2, 4, 6 of Algorithm 3 and steps 5 and 7 of Algorithm 1. Except for step 2 in Algorithm 3, all other steps can be solved in closed form. Subproblems solved in step 2 require \( O(d^2 T^2) \) optimization variables in the robust setting \( O(d^2 T^2) \) in the nominal setting \( O(d^2 T) \) in the nominal setting. The difference in complexity between the nominal and robust settings is consistent with the uncoupled scenario. Compared to the uncoupled scenario, additional computational burden is incurred by the consensus subroutine in Algorithm 3, which increases the total number of iterations. The consensus subroutine also induces increased communication between subsystems, as it requires local information exchange. However, this exchange is limited to a \( d \)-local region, resulting in small consensus problems that converge quickly, as we illustrate empirically in Section V. As with the original uncoupled algorithm, complexity of this new algorithm is determined by the size of the local neighborhood and does not increase with the size of the global network.

V. Simulation Experiments

Using examples, we demonstrate how adding terminal constraint and cost affects the performance of the DLMPC algorithm. We verify that introducing terminal constraint and cost produces the desired feasibility and stability. We empirically characterize the computational complexity of algorithms presented in previous sections and verify that complexity is independent of global network size for both offline and online algorithms.\(^6\)

\(^5\)By the assumptions in [15], \( \tilde{P} \) (and therefore \( \tilde{\mathcal{P}} \)) must induce linear inequalities in the robust setting. The only case where \( \tilde{P} \) may induce something other than linear inequalities is in the nominal setting.

\(^6\)Code to replicate these experiments is available at https://github.com/ unstable-zeros/dl-mpc-sls; this code makes use of the SLS toolbox at https://github.com/sls-caltech/sls-code, which includes ready-to-use MATLAB implementations of all algorithms presented in this paper and its companion paper [15].
Relative difference of the optimal cost obtained (i) with terminal set (damping), and (ii) with both terminal set and cost (yellow) compared to the optimal cost computed without terminal set and cost. Relative difference is obtained by taking the difference of the two costs and normalizing by the nonterminal-constrained cost. The difference obtained by adding the terminal set (indicated in pink) is on the order of $10^{-5}$ and is not visible on the plot.

### A. System Model

Simulations are performed on the system from our companion article [15]; a 2-D square mesh, where each node represents a two-state subsystem that follows linearized and discretized swing dynamics:

$$
\begin{align*}
\dot{\theta}(t+1) &= \sum_{j \in \text{in}(i)} [A]_{ij} \dot{\theta}(t) + \omega(t) + [B]_{i}[u]_i + [w]_i, \\
\omega(t+1) &= \sum_{j \in \text{in}(i)} [A]_{ij} \omega(t) + [B]_{i}[u]_i + [w]_i,
\end{align*}
$$

where $[\theta]_i$, $[\dot{\theta}]_i$, $[u]_i$, and $[w]_i$ are the phase angle deviation, frequency deviation, and control action of the controllable load of bus $i$. The dynamic matrices are

$$
[A]_{ii} = \begin{bmatrix} 1 & \Delta t \\
-\frac{1}{m_i} \Delta t & 1 - \frac{\Delta t}{m_i} \Delta t \end{bmatrix}, \quad [B]_{ij} = \begin{bmatrix} 0 & 0 \\
-\frac{k_{ij}}{m_i} \Delta t \end{bmatrix}
$$

and $[B]_{ii} = \begin{bmatrix} 0 & 1 \end{bmatrix}^\top$ for all $i$.

Connectivity among nodes is determined at random; each node connects to each of its neighbors with a 40% probability. The expected number of edges is $0.8 + n \times (n - 1)$. The parameters in bus $i$: $m_i^{-1}$ (inertia inverse), $d_i$ (damping), and $k_{ij}$ (coupling term) are randomly generated and uniformly distributed between $[0, 2]$, $[0.5, 1]$, and $[1, 1.5]$, respectively. We set the discretization step $\Delta t = 0.2$, and define $k_i := \sum_{j \in \text{in}(i)} k_{ij}$.

We study both the nominal setting and robust setting with uniformly distributed polytopic noise. The baseline parameter values are $d = 3$, $T = 5$, $N = 16$ ($4 \times 4$ grid). Unless otherwise specified, we start with a random-generated initial condition. We use a quadratic cost and polytopic constraints on both angle and frequency deviation, and impose upper and lower bounds.

### B. Performance

The addition of the terminal set and cost to the DLMPC algorithm introduces minimal conservatism in both nominal and robust settings. We study the two DLMPC cost for varying values of the time horizons for three different cases: 1) without terminal set and cost, 2) with terminal set, 3) with terminal set and terminal cost. Results are summarized in Fig. 1. Observe that the difference between the optimal cost across all three cases are negligible, indicating that our proposed terminal set and cost introduce no conservatism (while still providing the necessary theoretical guarantees).

Generally, the inclusion of the terminal set and cost introduce minimal change. In the vast majority of cases (e.g., all simulations from [15]), the DLMPC algorithm is feasible and stable even without a terminal set. This phenomenon has already been observed for the centralized case [1]. However, we also want to demonstrate how the terminal set and cost can make a difference, example, subsystem trajectories for the three different cases are shown in Fig. 2 for the nominal case and in the presence of polytopic disturbances. For these simulations only, we use a smaller ($N = 5$), more unstable ($m_i^{-1}$ between $[0, 16]$) system, extremely short time horizon ($T = 2$), and somewhat hand-crafted initial states and disturbances to obtain clearly visible differences between cases: without such instability, short time horizon, and hand-crafting, differences are generally tiny and not visible. In all cases, the centralized solution (computed...
via CVX [28]) coincides with the solution achieved by the DLMPC Algorithm 1, validating the optimality of the proposed algorithm. The effects of introducing terminal set and terminal cost are apparent and consistent with the theoretical results presented in this article.

### C. Computational Complexity

Simulations results verify the scalability of the proposed methods. We measure runtime\(^7\) while varying different network and problem parameters: locality \(d\), network size \(N\), and time horizon \(T\).\(^5\) We run five different simulations for each of the parameter combinations, using different realizations of the randomly chosen parameters to provide consistent runtime estimates.

First, we study the scalability of the offline Algorithm 2 to compute the terminal set; results are shown in Fig. 3. Consistent with theoretical analyses in Section IV, runtime does not increase with the size of the network; rather, it increases with the size of the neighborhood. As expected, computations for the robust set take slightly longer than for the nominal set, since the variables in the robust setting have greater dimension. Overall, synthesis times for both nominal and robust settings are extremely low, especially when a small locality size is used.

We also study how scalability of the DLMPC algorithm is affected when we impose a terminal set and terminal cost, and use Algorithm 3 to handle coupling. Results are shown in Fig. 4; this figure was generated using the same systems and parameters as Fig. 2 from our companion paper [15], allowing for direct comparison of online runtimes. The addition of the terminal set/cost slightly increases runtime, as expected. In the nominal case, runtime is increased from about \(10^{-3}\) s to \(10^{-1}\) s. In the case of polytopic disturbances, runtime is increased from 1–10 s to 10 s. Scalability is maintained; runtime barely increases with the size of the network. Overall, simulations indicate that the introduction of a terminal set and cost preserve scalability, minimally impact computational overhead and performance, and provide the desired guarantees.

### VI. Conclusion

In this article we provide theoretical guarantees for the closed-loop DLMPC approach, in both nominal and robust settings. In particular, we ensure recursive feasibility and stability by incorporating a terminal set and terminal cost. We also give guarantees for convergence of the algorithm. For the terminal set, we choose the maximal robust positive invariant set, which can be expressed compactly in the SLS parametrization. We introduce an algorithm to scalably compute this terminal set. We also provide the requisite modifications to the online DLMPC algorithm to accommodate local coupling induced by the terminal set and cost, as well as general coupling induced by process cost and constraints. All algorithms require only local information exchange, and enjoy computational complexity that is independent of the global system size. Although the cost and delays of the communication exchanges are beyond the scope of this article, if the communication delays and package dropouts can be modeled either as a polytopic or locally bounded disturbance, then the guarantees provided in this article for convergence, recursive feasibility and asymptotic stability still hold. The results in this article are the first to provide a distributed and localized computation of the maximal robust positive invariant control set and Lyapunov function of a large-scale system.

---

\(^7\)In online simulations, runtime is measured after the first iteration, so that all iterations for which runtime is measured are warm-started.

\(^5\)To increase network size, we vary the size of the grid over \(4 \times 4\) (32 states), \(6 \times 6\) (72 states), \(8 \times 8\) (128 states), and \(11 \times 11\) (242 states) grid sizes.

---

**REFERENCES**

[1] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
[2] D. Q. Mayne, M. M. Seron, and S. V. Rakovic, “Robust model predictive control of constrained linear systems with bounded disturbances,” *Automatica*, vol. 41, pp. 219–224, 2005.

[3] C. Langbort, R. S. Chandra, and R. D’Andrea, “Distributed control design for systems interconnected over an arbitrary graph,” *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1502–1519, Sep. 2004.

[4] A. Jokic and M. Lazar, “On decentralized stabilization of discrete-time nonlinear systems,” in *Proc. IEEE Amer. Control Conf.*, 2009, pp. 5777–5782.

[5] A. I. Zecevic and D. D. Siljak, *Control of Complex Systems*. New York, USA: Springer, 1988.

[6] B. T. Stewart, A. Venkat, J. RAWlings, S. Wright, and G. Pannocchia, “Cooperative distributed model predictive control,” *Syst. Control Lett.*, vol. 59, no. 8, pp. 460–469, 2010.

[7] J. M. Maestre, D. Muñoz de la Peña, E. F. CAMacho, and T. Alamo, “Distributed model predictive control based on agent negotiation,” *J. Process Control*, vol. 21, no. 5, pp. 685–697, 2011.

[8] C. Confe, C. N. Jones, M. Morari, and M. N. Zeilinger, “Distributed synthesis and stability of cooperative distributed model predictive control for linear systems,” *Automatica*, vol. 69, pp. 117–125, Jul. 2016.

[9] P. A. Trodden and J. M. Maestre, “Distributed predictive control with minimization of mutual disturbances,” *Automatica*, vol. 77, pp. 31–43, 2017.

[10] G. Darivianakis, A. Eichler, and J. Lygeros, “Distributed model predictive control for linear systems with adaptive terminal sets,” *IEEE Trans. Autom. Control*, vol. 65, no. 3, pp. 1044–1056, Mar. 2020.

[11] A. Aboudouina, A. Eichler, and J. Lygeros, “Distributed model predictive control with asymmetric adaptive terminal sets for the regulation of large-scale systems,” IFAC-PapersOnLine, vol. 53, no. 2, pp. 6899–6904, 2020.

[12] S. Muntwiler, K. P. Wabersich, A. Carron, and M. N. Zeilinger, “Distributed model predictive safety certification for learning-based control,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 5258–5262, 2020.

[13] Y. Wang and C. Manzie, “Robust distributed model predictive control for linear systems: Analysis and synthesis,” *Automatica*, vol. 137, 2022, Art. no. 110141.

[14] Y. Yu, Stutz, E. Z., Zhu, U. Rosolia, K. H. Johansson, and F. Borrelli, “Distributed learning model predictive control for linear systems,” in *Proc. IEEE Conf. Decis. Control*, 2020, pp. 4366–4373.

[15] C. Amo Alonso, J. S. Li, N. Matni, and J. Anderson, “Distributed and localized model predictive control. Part I: Synthesis and implementation,” *IEEE Trans. Control Netw. Syst.*, early access, Nov. 4, 2022, doi: 10.1109/TCNS.2022.3219770.

[16] J. Anderson, J. C. Doyle, S. H. Low, and N. Matni, “System level synthesis,” *Ann. Rev. Control*, vol. 47, pp. 364–393, 2019.

[17] F. Blanchini, “Set invariance in control,” *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[18] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.

[19] C. Amo Alonso and N. Matni, “Distributed and localized closed-loop model predictive control via system level synthesis,” in *Proc. IEEE Conf. Decis. Control*, 2020, pp. 5598–5605.

[20] D. Mayne, J. Rawlings, C. RAO, and P. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[21] J. Löfberg, “Minimax approaches to robust model predictive control,” Ph.D. dissertation Dept. Elect. Eng., Linköping Univ., Linköping, Sweden, 2003.

[22] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: New York, NY, USA: Cambridge Univ. Press, 2004.

[23] K. Rokade and R. K. Kalaimani, “Distributed ADMM over directed networks,” *2020, arXiv:2010.10421*.

[24] C. Amo Alonso, D. Ho, and J. Maestre, “Distributed linear quadratic regulator robust to communication dropouts,” IFAC-PapersOnLine, vol. 53, no. 2, pp. 3072–3078, 2020.

[25] I. Yu, Y. S. Wang, and J. Anderson, “Localized and distributed $H_2$ state feedback control,” in *Proc. IEEE Amer. Control Conf.*, 2021, pp. 2732–2738.

[26] E. Gilbert and K. Tan, “Linear systems with state and control constraints: The theory and applications of maximal output admissible sets,” *IEEE Trans. Autom. Control*, vol. 36, no. 9, pp. 1008–1020, Sep. 1991.

[27] G. Costantini, R. Rostami, and D. Gorges, “Decomposition methods for distributed quadratic programming with application to distributed model predictive control,” in *Proc. IEEE Proc. Annu. Allerton Conf. Commun., Control, Comput.*, 2018, pp. 943–950.

[28] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” Mar. 2014. [Online]. Available: http://cvxr.com/cvx

**Carmen Amo Alonso** received the B.Sc. degree in aerospace engineering from the Politecnico University of Madrid, Madrid, Spain, in 2016, and the M.Sc. degree in space engineering, in 2017, from the California Institute of Technology, Pasadena, CA, USA, where she is currently working toward the Ph.D. degree in control and dynamical systems.

Her research interests include distributed optimal control, convex optimization, and parallel programming.

Ms. Amo Alonso is a Fellow of AI4Science Amazon/Caltech.

**Jing Shuang Li** (Student Member, IEEE) received the B.A.Sc. degree in engineering science from the University of Toronto, Toronto, ON, Canada, in 2018. She is currently working toward the Ph.D. degree in control and dynamical systems with the California Institute of Technology, Pasadena, CA, USA.

Her research interests include distributed control and its applications to sensorimotor and biological modeling.

**Nikolai Matni** (Senior Member, IEEE) received the Ph.D. degree in control and dynamical systems from Caltech, Pasadena, CA, USA, in 2016.

He is an Assistant Professor with the Department of Electrical and Systems Engineering, University of Pennsylvania (Penn), Philadelphia, PA, USA. Prior to joining Penn, he was a Postdoctoral Scholar in EECS with UC Berkeley, Berkeley, CA, USA. He also was a Postdoctoral Scholar in computing and mathematical sciences with Caltech.

Dr. Nikolai was the recipient of the NSF CAREER Award in 2021, a Google Research Scholar Award in 2021, and the 2021 IEEE CSS George S. Axelby Award.

**James Anderson** (Senior Member, IEEE) received the DPhil (Ph.D.) and MA degree from the University of Oxford, Oxford, U.K., in 2012, both in systems engineering.

He is an Assistant Professor with the Department of Electrical Engineering, Columbia University, New York, NY, USA, where he is also a member of the Data Science Institute. From 2016 to 2019, he was a Senior Postdoctoral Scholar in computing and mathematical sciences with Caltech, Pasadena, CA, USA. Prior to Caltech, he held a Junior Research Fellowship at the University of Oxford. His research interests include optimal and distributed control, convex optimization, and scientific computing with Big Data.