2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Plug and Play Distributed Model Predictive Control with Dynamic Coupling: A Randomized Primal-Dual Proximal Algorithm

Puya Latafat, Alberto Bemporad, Panagiotis Patrinos

Abstract— **This paper proposes an algorithm for distributed model predictive control that is based on a primal-dual proximal algorithm developed recently by two of the authors. The proposed scheme does not require strong convexity, involves one round of communication at every iteration and is fully distributed. In fact, both the iterations and the stepsizes are computed using only local information. This allows a plug and play implementation where addition or removal of a subsystem only affects the neighboring nodes without the need for global coordination. The proposed scheme enjoys a linear convergence rate. In addition, we provide a randomized variant of the algorithm in which at every iteration subsystems wake up randomly independent of one another. Numerical simulations are performed for the frequency control problem in a power network, demonstrating the attractive performance of the new scheme.**

I. Introduction

This paper considers distributed model predictive control (DMPC) of a network of m dynamically coupled linear systems. For $i = 1, \ldots, m$, the dynamics of system i is of the form

$$x_i(k+1) = \sum_{j=1}^{m} \Phi_{ij} x_j(k) + \Delta_{ij} u_j(k),$$

with $x_i(k) \in \mathbb{R}^{s_i}$, $u_i(k) \in \mathbb{R}^{t_i}$, subject to local state and input constraints. The structure of the network is defined by matrices Φ_{ij} and Δ_{ij}. System j affects i if either one of Φ_{ij}, Δ_{ij} is nonzero. It is natural to assume that two systems can communicate if either one of them affects the dynamics of the other, in which case we say that they are neighbors. However, the systems need not be aware of the global structure of the network, or even existence of systems beyond their neighbors.

DMPC formulations considered in the literature vary depending on the nature of the coupling and can be grouped in two general categories. In applications such as formation control where the systems are physically separate but share a common goal the DMPC problem involves coupling cost or constraints without dynamic coupling [1]–[3]. The second category involves DMPC problems with dynamic coupling with applications ranging from smart grids, sensor networks, water networks to transportation systems, and has been studied by many authors [4]–[9]. This paper is focused on the second category. We note that in our setting it is straightforward to extend the proposed setup to include coupling in cost and constraint between neighbors, however, this leads to complicated notation and has been avoided for the sake of clarity. Furthermore, this work is not concerned with the stability of the closed-loop system and looks at the DMPC problem from the optimization point of view.

A popular approach for solving the DMPC problem is to derive distributed algorithms using dual decomposition. Many authors have considered solving the dual problem using the proximal gradient method, the alternating direction method of multipliers (ADMM) or their variants [4], [6]–[8], [10]. These approaches are preferred to subgradient methods given that they allow constant stepsizes. Algorithms that are based on proximal gradient or its accelerated variants require the cost function to be strongly convex. Another common issue is the need for centralized computations for selecting the stepsizes. This is a major drawback that can hinder the implementation especially in applications where the network structure is subject to change. For example, applying proximal gradient requires the stepsize to be bounded by the inverse of the Lipschitz constant associated to the dual function [4]. In [11] a metric for Lipschitz continuity is used which requires solving a semidefinite program (SDP) globally. In [6] the authors provide a distributed method for selecting the metric that involves solving a series of local SDPs. Another recent work that involves distributed stepsize selection is [7] where the Lagrangian minimization step is modified with regularization terms. Each iteration in [6] and [7] involve a local inner minimization step the result of which is required by the neighbors, *i.e.*, each iteration involves two rounds of communication.

In contrast to the dual decomposition approach our proposed algorithm is the result of applying a new primal-dual proximal algorithm [12] to the primal problem. The aforementioned algorithm is based on a new operator splitting technique [13] and is specifically tailored for distributed applications. The main contributions of this paper are summarized below:

- The new algorithm is *fully*-distributed, involves simple computations for each subsystem without any inner loops, and requires one round of communication per update. At every iteration active subsystems perform local updates, communicate the necessary vectors to their neighbors, and

Puya Latafat[1,2]; Email: puya.latafat@{kuleuven.be,imtlucca.it}
Alberto Bemporad[2]; Email: alberto.bemporad@imtlucca.it
Panagiotis Patrinos[1]; Email: panos.patrinos@esat.kuleuven.be

This work was supported by the Research Foundation Flanders (FWO) PhD fellowship 1196818N; FWO research projects G086518N and G086318N; KU Leuven internal funding StG/15/043; Fonds de la Recherche Scientifique – FNRS and the Fonds Wetenschappelijk Onderzoek – Vlaanderen under EOS Project no 30468160 (SeLMA)

[1] KU Leuven, Department of Electrical Engineering (ESAT-STADIUS), Kasteelpark Arenberg 10, 3001 Leuven-Heverlee, Belgium.
[2] IMT School for Advanced Studies Lucca, Piazza San Francesco 19, 55100 Lucca, Italy.

go idle. The algorithm is presented in two forms: The synchronous case where all of the systems are active at every iteration, and the asynchronous case where subsystems are activated at random independently of one another.

- The stepsize of each subsystem is selected locally through a simple rule (cf. Assumption 2*(iii)*). Therefore, any modification to the network structure would only affect the neighboring subsystems.
- The cost function must be convex but not necessarily strongly convex.
- The algorithm possesses linear convergence rate when the local input and state constraints are polyhedral sets, a common scenario.

II. Problem Setup

We consider a distributed model predictive control problem with m dynamically coupled subsystems. We use an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to model the interaction between subsystems/agents. Each node $i \in \mathcal{V}$ is associated with a *subsystem*, maintains its own local variables, and can communicate with its neighbors. The goal is to solve the global model predictive control problem with only local exchange of information between neighbors.

Let Φ_{ij} and Δ_{ij} denote the state transition and input matrices from subsystem j to i. For all $i \in \mathcal{V}$, the *in-neighbor* and *out-neighbor* sets are defined as follows:

$$\begin{aligned}\mathcal{N}_i^{\text{in}} &= \{j \in \mathcal{V}\setminus\{i\} | \Phi_{ij} \neq 0 \text{ or } \Delta_{ij} \neq 0\},\\ \mathcal{N}_i^{\text{out}} &= \{j \in \mathcal{V}\setminus\{i\} | \Phi_{ji} \neq 0 \text{ or } \Delta_{ji} \neq 0\},\end{aligned}$$

and the neighborhood set is defined by $\mathcal{N}_i = \mathcal{N}_i^{\text{out}} \cup \mathcal{N}_i^{\text{in}}$, *i.e.*, the edge $(i,j) \in \mathcal{E}$ exists if $j \in \mathcal{N}_i$. The DMPC problem is written in the following standard form:

$$\begin{aligned}\text{minimize } & \frac{1}{2}\sum_{i=1}^{m}\Big(\sum_{k=1}^{N} x_i(k)^\top Q_i^k x_i(k) + \sum_{k=0}^{N-1} u_i(k)^\top R_i^k u_i(k)\Big)\\ \text{subject to } & x_i(k+1) = \sum_{j\in\mathcal{N}_i^{\text{in}}\cup\{i\}} \Phi_{ij}x_j(k) + \Delta_{ij}u_j(k) \quad \text{(1a)}\\ & u_i(k) \in \mathcal{U}_i, \quad \text{for } k = 0,\ldots,N-1,\\ & x_i(k) \in \mathcal{X}_i, \quad \text{for } k = 1,\ldots,N-1,\\ & x_i(N) \in \mathcal{X}_i^f\\ & \text{for all } i = 1,\ldots,m\end{aligned}$$

where $x_i(0)$ is given, $x_i(k) \in \mathbb{R}^{s_i}$ and $u_i(k) \in \mathbb{R}^{t_i}$ denote the state and input variables of subsystem i at time k.

Note that the separable quadratic cost function is used for clarity of exposition. It may be replaced by any Lipschitz differentiable function without requiring the subsystems to solve inner minimizations (cf. Remark 1). Furthermore, it is straightforward to modify our analysis to allow coupling between neighbors.

Throughout the paper the following assumptions hold:

Assumption 1. For $i = 1,\ldots,m$:

(i) Input and state constraint sets $\mathcal{X}_i, \mathcal{X}_i^f \subseteq \mathbb{R}^{s_i}$ and $\mathcal{U}_i \subseteq \mathbb{R}^{t_i}$ are nonempty, closed, and convex.

(ii) The cost matrices Q_i^k and R_i^k are positive semidefinite.

(iii) The graph $\mathcal{G}$ is connected.

(iv) The DMPC problem admits a solution. Moreover, for $i = 1,\ldots,m$ there exists $x_i(k) \in \operatorname{ri}\mathcal{X}_i$ for $k = 1,\ldots,N-1$, $x_i(N) \in \operatorname{ri}\mathcal{X}_i^f$, and $u_i(k) \in \operatorname{ri}\mathcal{U}_i$ for $k = 0,\ldots,N-1$ such that the linear dynamics (1a) are satisfied ($\operatorname{ri} C$ denotes the relative interior of the set C).

The strict feasibility enforced in Assumption 1*(iv)* ensures that strong duality holds, and can be dropped whenever the constraint sets are polyhedral [14, Corollary 31.2.1].

For $i = 1,\ldots,m$ define

$$z_i = \big(x_i(1),\cdots,x_i(N),u_i(0),\cdots,u_i(N-1)\big) \in \mathbb{R}^{r_i},$$

where $r_i = N(s_i + t_i)$. The quadratic cost function can be written as $\frac{1}{2}\sum_{i=1}^m z_i^\top G_i z_i$ where $G_i = \operatorname{blkdiag}(Q_i^1,\ldots,Q_i^N,R_i^0,\ldots,R_i^{N-1})$. The dynamics can be expressed as:

$$\sum_{j\in\mathcal{N}_i^{\text{in}}\cup\{i\}} L_{ij}z_j = \sum_{j\in\mathcal{N}_i^{\text{in}}\cup\{i\}} b_{ij}x_j(0), \quad \text{for } i = 1,\cdots,m,$$

where L_{ij} and b_{ij} are appropriate linear mappings [11]. With these definitions the distributed MPC problem becomes

$$\begin{aligned}\text{minimize } & \frac{1}{2}\sum_{i=1}^{m} z_i^\top G_i z_i && \text{(2a)}\\ \text{subject to } & \sum_{j\in\mathcal{N}_i^{\text{in}}\cup\{i\}} L_{ij}z_j = b_i, \quad i = 1,\cdots,m && \text{(2b)}\\ & z_i \in \mathcal{Z}_i, \quad i = 1,\cdots,m && \text{(2c)}\end{aligned}$$

where $b_i = \sum_{j\in\mathcal{N}_i^{\text{in}}\cup\{i\}} b_{ij}x_j(0)$, and the constraint sets $\mathcal{Z}_i$ denote the product of local input and state constraint sets:

$$\mathcal{Z}_i = \underbrace{\mathcal{X}_i \times \ldots \times \mathcal{X}_i}_{N-1} \times \mathcal{X}_i^f \times \underbrace{\mathcal{U}_i \times \ldots \times \mathcal{U}_i}_{N}.$$

III. A Primal-Dual Algorithm for DMPC

Our goal is to solve (2) in a fully distributed fashion while keeping the number of communications to a minimum. For each subsystem that affects i, *i.e.*, $j \in \mathcal{N}_i^{\text{in}}$, we introduce a local variable z_{ij}, that can be seen as the estimate of z_j kept locally by agent i. For notation consistency, self-variables z_i are hereafter denoted by z_{ii}. We write the equivalent optimization problem:

$$\begin{aligned}\text{minimize } & \frac{1}{2}\sum_{i=1}^{m} z_{ii}^\top G_i z_{ii} && \text{(3a)}\\ \text{subject to } & \sum_{j\in\mathcal{N}_i^{\text{in}}\cup\{i\}} L_{ij}z_{ij} = b_i, \quad i = 1,\ldots,m && \text{(3b)}\\ & z_{ii} \in \mathcal{Z}_i, \quad i = 1,\ldots,m && \text{(3c)}\\ & z_{ij} = z_{jj}, \quad i = 1,\ldots,m \text{ and } j \in \mathcal{N}_i^{\text{in}} && \text{(3d)}\end{aligned}$$

For $i \in \mathcal{V}$, let $n_i = \sum_{j\in\mathcal{N}_i^{\text{in}}\cup\{i\}} r_j$ and define:

$$z_{\mathcal{N}_i} = (z_{ij})_{j\in\mathcal{N}_i^{\text{in}}\cup\{i\}} \in \mathbb{R}^{n_i}, \ L_i = [L_{ij}]_{j\in\mathcal{N}_i^{\text{in}}\cup\{i\}} \in \mathbb{R}^{Ns_i\times n_i}$$

The set of points satisfying the linear constraint is given by:

$$\mathcal{D}_i = \{z \in \mathbb{R}^{n_i} | L_i z = b_i\}.$$

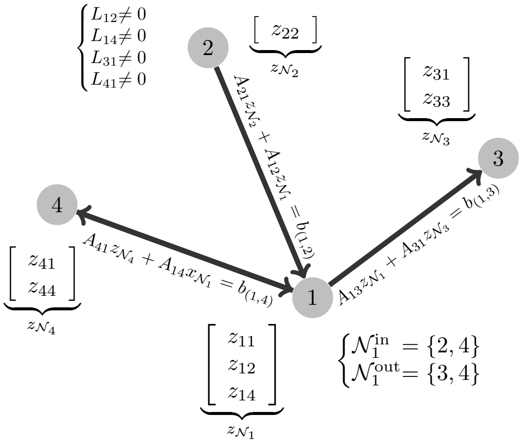
$L_{12}\neq 0$
$L_{14}\neq 0$
$L_{31}\neq 0$
$L_{41}\neq 0$
2
z_{22}
$z_{\mathcal{N}_2}$
z_{31}
z_{33}
$z_{\mathcal{N}_3}$
3
4
$A_{21}z_{\mathcal{N}_2} + A_{12}z_{\mathcal{N}_1} = b_{(1,2)}$
$A_{41}z_{\mathcal{N}_4} + A_{14}x_{\mathcal{N}_1} = b_{(1,4)}$
$A_{13}z_{\mathcal{N}_1} + A_{31}z_{\mathcal{N}_3} = b_{(1,3)}$
z_{41}
z_{44}
$z_{\mathcal{N}_4}$
1
z_{11}
z_{12}
z_{14}
$z_{\mathcal{N}_1}$
$\mathcal{N}_1^{\text{in}} = \{2,4\}$
$\mathcal{N}_1^{\text{out}} = \{3,4\}$

Fig. 1. Dynamic coupling in the DMPC problem

We note that the variables are stacked in ascending order (index-wise). For example, consider the neighborhood relation described in Figure 1. Subsystem 1 is affected by subsystems 2 and 4, therefore, $z_{\mathcal{N}_1} = (z_{11}, z_{12}, z_{14})$. Our proposed algorithm is a primal-dual scheme. Therefore, in addition to primal variables each system holds dual variables. For each $i \in \mathcal{V}$ we introduce two sets of dual variables: the node variable $y_i \in \mathbb{R}^{r_i}$ and the edge variables $w_{ij,i} \in \mathbb{R}^{r_i}$ for $j \in \mathcal{N}_i^{\text{out}}$, and $w_{ji,i} \in \mathbb{R}^{r_j}$ for $j \in \mathcal{N}_i^{\text{in}}$. The first argument of the subscript denotes the edge relation and the second the ownership of the variable, *i.e.*, if system i affects j, then i and j will keep $w_{ij,i}$ and $w_{ij,j}$ respectively.

Let $E_i \in \mathbb{R}^{r_i \times n_i}$ be a linear mapping such that $E_i z_{\mathcal{N}_i} = z_{ii}$. Define

$$g_i(z_{\mathcal{N}_i}) = \delta_{\mathcal{D}_i}(z_{\mathcal{N}_i}) + \tfrac{1}{2} z_{\mathcal{N}_i}^\top E_i^\top G_i E_i z_{\mathcal{N}_i}, \quad h_i = \delta_{\mathcal{Z}_i}, \tag{4}$$

where δ_X denotes the indicator function of a closed nonempty convex set, X. Problem (3) becomes

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^m g_i(z_{\mathcal{N}_i}) + h_i(E_i z_{\mathcal{N}_i}) && \text{(5a)}\\ &\text{subject to} && A_{ij}z_{\mathcal{N}_i} + A_{ji}z_{\mathcal{N}_j} = 0 \qquad (i,j) \in \mathcal{E} && \text{(5b)} \end{aligned}$$

where $A_{ij} \in \mathbb{R}^{l_{(i,j)} \times n_i}$ is defined based on the neighborhood relation as follows

$$A_{ij}z_{\mathcal{N}_i} = \begin{cases} z_{ii} & j \notin \mathcal{N}_i^{\text{in}},\ j \in \mathcal{N}_i^{\text{out}} \\ -z_{ij} & j \in \mathcal{N}_i^{\text{in}},\ j \notin \mathcal{N}_i^{\text{out}} \\ (z_{ii}, -z_{ij}) & j \in \mathcal{N}_i^{\text{in}},\ j \in \mathcal{N}_i^{\text{out}}. \end{cases} \tag{6}$$

Notice that depending on the neighborhood relation $l_{(i,j)}$ is either equal to r_i, r_j or $r_i + r_j$.

A primal-dual algorithm was introduced in [15] for problems of the form (5) with consensus constraint. However, a consensus constraint can not capture the coupling in the DMPC problem depicted in Figure 1. Another drawback of the aforementioned work is that the stepsize selection requires global coordination. Our analysis here is different from that work and is based on [12]. In Section IV we describe how [12, Algorithm 3] is applied to the DMPC problem to derive Algorithm 1.

Before proceeding with the algorithm recall the definition of Moreau's proximal mapping [16]. For a positive definite matrix V and proper closed convex function $q : \mathbb{R}^n \mapsto \overline{\mathbb{R}}$, the *proximal mapping* relative to $\|\cdot\|_V$ is defined as:

$$\operatorname{prox}_q^V(x) := \operatorname*{argmin}_{z \in \mathbb{R}^n}\{q(z) + \tfrac{1}{2}\|x - z\|_V^2\}.$$

Our proposed distributed scheme is summarized in Algorithm 1. It involves two versions. In the synchronous case at each iteration all systems perform their local updates and broadcast the result to the relevant neighbors. In the asynchronous case each system wakes up randomly independent of other systems, *i.e.*, there may be several active systems at each iteration. The stepsizes appearing in Algorithm 1 should satisfy the following assumption:

Assumption 2 (Stepsizes in Algorithm 1)**.**

(i) (node stepsizes) Subsystem i keeps two positive stepsizes σ_i, τ_i associated to h_i and g_i, respectively.

(ii) (edge stepsizes) For each edge $(i,j) \in \mathcal{E}$ we associate a positive stepsize $\kappa_{(i,j)}$ that is shared between system i and j.

(iii) (convergence condition) The stepsizes satisfy the following local condition consensus

$$\tau_i < \frac{1}{\max\{\sum_{j \in \mathcal{N}_i^{\text{out}}} \kappa_{(i,j)} + \sigma_i, (\kappa_{(i,j)})_{j \in \mathcal{N}_i^{\text{in}}}\}}. \tag{7}$$

The dual updates for y_i in Algorithm 1 require projection onto the set $\mathcal{Z}_i$ which can often be performed efficiently, *e.g.* for boxes, halfspaces, norm balls. The primal updates are compactly written as $z_{\mathcal{N}_i} = \operatorname{prox}_{\tau_i g_i}(c)$ where $c = (c_{ij})_{j \in \mathcal{N}_i^{\text{in}} \cup \{i\}}$ is given by

$$c_{ii} = z_{ii} - \tau_i\big(\bar{y}_i + \sum_{j \in \mathcal{N}_i^{\text{out}}} \bar{w}_{ij,i}\big), \tag{8a}$$

$$c_{ij} = z_{ij} + \tau_i \bar{w}_{ji,i}, \qquad \text{for all } j \in \mathcal{N}_i^{\text{in}}. \tag{8b}$$

The proximal mapping $\operatorname{prox}_{\tau_i g_i}(c)$ involves the minimization of a strongly convex quadratic function over an affine subspace:

$$\begin{aligned} &\underset{z}{\text{minimize}} && \tfrac{1}{2}z^\top(E_i^\top G_i E_i + \tfrac{1}{\tau_i}I_{n_i})z - \tfrac{1}{\tau_i}c^\top z && \text{(9a)}\\ &\text{subject to} && L_i z = b_i, && \text{(9b)} \end{aligned}$$

and can be evaluated efficiently through solving the linear system defining its KKT optimality conditions. We stress that the matrix of the linear system is constant throughout iterations, and needs to be factored only once. Consequently, the evaluation of the primal step at every iteration amounts to forward and backward substitution steps [17, Section III.C].

IV. Deriving the Algorithm and Convergence Results

In this section we detail the steps of applying [12, Algorithm 1 and 2] to the DMPC problem.

Let $z = (z_{\mathcal{N}_1}, \ldots, z_{\mathcal{N}_m})$ and define the linear operator

$$N_{(i,j)} : z \mapsto (A_{ij}z_{\mathcal{N}_i}, A_{ji}z_{\mathcal{N}_j}).$$

Algorithm 1 Synchronous & asynchronous distributed primal-dual algorithm for DMPC

```
Inputs: $\sigma_i > 0$, $\tau_i > 0$ for $i = 1, \dots, m$, and $\kappa_{(i,j)} > 0$ for $(i,j) \in \mathcal{E}$.
for $k = 0, \dots$ do
    I: Synchronous version
        for all systems $i = 1, \dots, m$ do
    II: Asynchronous version
        draw r.v. $\epsilon_i^k$ according to $\mathbb{P}(\epsilon_i^0 = 1) = p_i > 0$
        for all systems $i$ with $\epsilon_i^k = 1$ do
    Local updates:
        $\bar{w}_{ij,i}^k = \frac{1}{2}\left(w_{ij,i}^k + w_{ij,j}^k\right) + \frac{\kappa_{(i,j)}}{2}\left(z_{ii}^k - z_{ji}^k\right)$, for all $j \in \mathcal{N}_i^{\text{out}}$
        $\bar{w}_{ji,i}^k = \frac{1}{2}\left(w_{ji,i}^k + w_{ji,j}^k\right) + \frac{\kappa_{(i,j)}}{2}\left(z_{jj}^k - z_{ij}^k\right)$, for all $j \in \mathcal{N}_i^{\text{in}}$
        $\bar{y}_i^k = y_i^k + \sigma_i z_{ii}^k - \sigma_i \mathcal{P}_{\mathcal{Z}_i}\left(\sigma_i^{-1} y_i^k + z_{ii}^k\right)$
        $z_{\mathcal{N}_i}^{k+1} = \operatorname{prox}_{\tau_i g_i}(c)$ is given by (8) and (9).
        $y_i^{k+1} = \bar{y}_i^k + \sigma_i(\bar{z}_{ii}^k - z_{ii}^k)$
        $w_{ij,i}^{k+1} = \bar{w}_{ij,i}^k + \kappa_{(i,j)}(\bar{z}_{ii}^k - z_{ii}^k)$, for all $j \in \mathcal{N}_i^{\text{out}}$
        $w_{ji,i}^{k+1} = \bar{w}_{ji,i}^k - \kappa_{(i,j)}(\bar{z}_{ij}^k - z_{ij}^k)$, for all $j \in \mathcal{N}_i^{\text{in}}$
    Broadcast of information:
        Send $z_{ii}^{k+1}, w_{ij,i}^{k+1}$ to $j \in \mathcal{N}_i^{\text{out}}$, and $z_{ij}^{k+1}, w_{ji,i}^{k+1}$ to $j \in \mathcal{N}_i^{\text{in}}$
```

The edge constraints (5b) can be equivalently formulated in the cost as $\sum_{i=1}^m \delta_{C_{(i,j)}}\left(N_{(i,j)} z\right)$, where $C_{(i,j)} = \{(z_1, z_2) \in \mathbb{R}^{2l_{(i,j)}} \mid z_1 + z_2 = 0\}$. Consequently, (5) can be formulated in the form of unconstrained optimization:

$$\text{minimize} \sum_{i=1}^m \left(g_i(z_{\mathcal{N}_i}) + h_i(E_i z_{\mathcal{N}_i})\right) + \sum_{(i,j)\in\mathcal{E}} \delta_{C_{(i,j)}}\left(N_{(i,j)} z\right)$$

In order to formulate the dual problem we introduce two sets of dual variables, $y_i \in \mathbb{R}^{r_i}$ and $w_{(i,j)} \in \mathbb{R}^{2l_{(i,j)}}$. The former corresponds to node and the latter to edge constraints. The edge variable $w_{(i,j)}$ consists of two blocks, $w_{(i,j)} = (w_{(i,j),i}, w_{(i,j),j})$, *i.e.*, we consider two dual variables for each constraint, where $w_{(i,j),i} \in \mathbb{R}^{l_{(i,j)}}$ is maintained by agent i and $w_{(j,i),j} \in \mathbb{R}^{l_{(i,j)}}$ by agent j. Notice that the edge variable $w_{(i,j),i}$ itself consists of either one or two blocks: $w_{(i,j),i} = (w_{ij,i}, w_{ji,i})$, where $w_{ij,i}$ and $w_{ji,i}$ are present when $j \in \mathcal{N}_i^{\text{out}}$ and $j \in \mathcal{N}_i^{\text{in}}$, respectively.

For clarity of exposition we rewrite the problem with compact notation. We use N without any subscript to denote the stacked linear mapping $\mathsf{N} = (N_{(i,j)})_{(i,j)\in\mathcal{E}}$, and $C = \bigtimes_{(i,j)\in\mathcal{E}} C_{(i,j)}$. The transpose of N is given by

$$\mathsf{N}^\top : (w_{(i,j)})_{(i,j)\in\mathcal{E}} \mapsto \tilde{\mathsf{z}} = \sum_{(i,j)\in\mathcal{E}} \mathsf{N}_{(i,j)}^\top w_{(i,j)},$$

with $\tilde{\mathsf{z}}_i = \sum_{j\in\mathcal{N}_i} A_{ij}^\top w_{(i,j),i}$. Furthermore, set $E = \operatorname{blkdiag}(E_1, \dots, E_m)$ and define $Lz = (Ez, \mathsf{N}z) =: (\tilde{y}, \tilde{w}) \in \mathbb{R}^{n_d}$, where $n_d = \sum_{(i,j)\in\mathcal{E}} 2l_{(i,j)} + \sum_{i=1}^m r_i$. Set $g(z) = \sum_{i=1}^m g_i(z_{\mathcal{N}_i})$, $h(\tilde{y}, \tilde{w}) = \tilde{h}(\tilde{y}) + \delta_C(\tilde{w})$, where $\tilde{h}(\tilde{y}) = \sum_{i=1}^m h_i(\tilde{y}_i)$. Then, problem (5) can be casted as

$$\text{minimize } g(z) + h(Lz). \tag{10}$$

Problem (10) may be solved by a range of primal-dual algorithms resulting in the full splitting of the nonsmooth functions and the linear mapping [13, Algorithm 3 and Figure 1]. In this work our goal is to derive algorithms in which: i) both the iterates and the stepsizes are computed locally, ii) involve one round of communication per iteration, iii) allow block coordinate updates. An ideal candidate for this purpose is the primal-dual algorithm introduced in [12, Algorithm 1]. In particular, the sequence generated by the algorithm is Fejér monotone with respect to $\|\cdot\|_S$ where S is a block diagonal positive definite matrix. This is not the case in other closely related primal-dual algorithms such as the Chambolle-Pock algorithm [18], where the linear mapping L appears as the off-diagonal element of S (see [12, Section II]).

Remark 1. In (4) the quadratic terms were captured by nonsmooth functions g_i. Our scheme requires calculating the proximal mapping of g_i which translates to solving the quadratic over affine minimization (9). Alternatively, one can model the quadratic cost functions using a third smooth term in (10) (see [12, Algorithm 1]). This would result in a gradient step and a projection onto the set $\mathcal{D}_i$ in place of a quadratic over affine minimization. Hence, it is possible to use general convex Lipschitz differentiable functions as cost in the DMPC problem. In that case the Lipschitz constant of the smooth term would affect the stepsizes. $\diamondsuit$

In order to represent the algorithm compactly we define the following set of diagonal matrices:

$$\begin{aligned} W &= \operatorname{blkdiag}\left((\kappa_{(i,j)} I_{2l_{(i,j)}})_{(i,j)\in\mathcal{E}}\right), \\ \Sigma &= \operatorname{blkdiag}(\sigma_1 I_{r_1}, \dots, \sigma_m I_{r_m}), \\ \Gamma &= \operatorname{blkdiag}(\tau_1 I_{n_1}, \dots, \tau_m I_{n_m}). \end{aligned}$$

Notice that $\kappa_{(i,j)}$ is repeated twice, *i.e.*, once for every node sharing the edge.

Let $v = (y, w, z)$, and define the operator T

$$Tv = (\bar{y} + \Sigma E(\bar{z} - z), \bar{w} + WN(\bar{z} - z), \bar{z}),$$

where

$$\bar{y} = \operatorname{prox}_{\tilde{h}^*}^{\Sigma^{-1}}(y + \Sigma E z) \tag{11a}$$

$$\bar{w} = \operatorname{prox}_{\delta_C^*}^{W^{-1}}(w + WNz) \tag{11b}$$

$$\bar{z} = \operatorname{prox}_g^{\Gamma^{-1}}(z - \Gamma E^\top \bar{y} - \Gamma N^\top \bar{w}). \tag{11c}$$

Then [12, Algorithm 1] can be represented as the fixed-point iteration $v^{k+1} = Tv^k$. This iteration is amenable to *block coordinate* (BC) updates. A general BC scheme was proposed in [12, Algorithm 2]. Our focus here is on the case where each coordinate has an independent probability to be active. Briefly put, the BC scheme is represented as

$$z^{k+1} = \sum_{i=1}^{m} \epsilon_i^k U_i(Tz^k),$$

where U_i are diagonal matrices with zero and one diagonal elements, and are used to select the coordinates, while $\epsilon_i^k \in \{0,1\}^m$ encodes if a coordinate i is updated at iteration k. The matrices U_i are assumed to be disjoint and $\sum_{i=1}^m U_i = I$, where I is the identity matrix of appropriate dimensions. The partitioning described in this section satisfies these requirements, *i.e.*, for $i = 1, \ldots, m$, the matrix U_i selects $z_{\mathcal{N}_i}, y_i$ and $w_{(i,j),i}$ for $j \in \mathcal{N}_i$.

Since $\tilde{h}$ in (11a) is separable, using (4) and the Moreau identity [16], we have that for $i = 1, \ldots, m$

$$\bar{y}_i = y_i + \sigma_i z_{ii} - \mathcal{P}_{\mathcal{Z}_i}(y_i + \sigma_i z_{ii}).$$

Furthermore, the projection onto $C_{(i,j)}$ is given by $\mathcal{P}_{C_{(i,j)}}(w_1, w_2) = \frac{1}{2}(w_1 - w_2, -w_1 + w_2)$. Therefore, (11b) yields the updates for the edge variables in Algorithm 1. The $\bar{z}$ in (11c) can be evaluated as follows: For each $i \in \mathcal{V}$

$$\bar{z}_{\mathcal{N}_i} = \mathrm{prox}_{\tau_i g_i}\Big(z_{\mathcal{N}_i} - \tau_i E_i^\top \bar{y}_i - \tau_i \sum_{j \in \mathcal{N}_i} A_{ij}^\top \bar{w}_{(i,j),i}\Big).$$

Therefore, the primal update is carried out by solving (9) where c is given by (8). Finally, evaluation of the operator T requires matrix vector products and straightforward substitution of the involved matrices yields Algorithm 1.

Next theorem summarizes the convergence results for Algorithm 1. The proof is omitted here and the interested reader is referred to [12, Lemma 3 and Theorem 6,7].

Theorem 1. *Let Assumptions 1 and 2 hold. Consider the stacked vectors* $z = (z_{\mathcal{N}_1}, \ldots, z_{\mathcal{N}_m})$, $y = (y_1, \ldots, y_m)$, $w = (w_{(i,j)})_{(i,j)\in\mathcal{E}}$. *Then, in the case of synchronous updates,* $(v^k)_{k\in\mathbb{N}} = (y^k, w^k, z^k)_{k\in\mathbb{N}}$ *generated by Algorithm 1 converges to some* $v^\star$, *and in the case of asynchronous updates it converges almost surely to some* $v^\star$*-valued random variable, where* $v^\star$ *is a primal-dual solution to* (5)*. In particular,* $(z_{11}^k, \ldots, z_{mm}^k)_{k\in\mathbb{N}}$ *converge to a solution of the DMPC problem* (1)*. If in addition,* $\mathcal{X}_i$, $\mathcal{X}^f$ *and* $\mathcal{U}_i$ *are polyhedral sets then in the synchronous case the distance from the primal-dual solution set converges Q-linearly to zero.*

V. Numerical Simulations

In this section, as a benchmark example we consider the problem of frequency control in power networks [19]. The network consists of power generation areas with the goal of maintaining nominal frequency levels despite changes in load and network configuration. The approach in [19] is based on modeling the dynamic coupling as disturbance. Clearly, this could lead to conservative control actions. In contrast our method solves the exact global optimization constrained by the dynamics through distributed computation and communication with the neighbors.

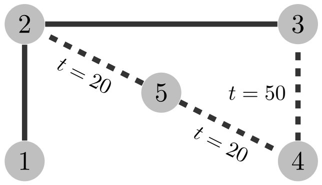

Fig. 2. Network structure in the DMPC problem for scenario 1: system 5 is added at $t = 20$ and system 4 is disconnected at $t = 50$.

Each system consists of four states $x_i = (\Delta\theta_i, \Delta\omega_i, \Delta P_{\mathrm{m}_i}, \Delta P_{\mathrm{v}_i})$ and one control input $u_i = \Delta P_{\mathrm{ref}_i}$. The continuous-time LTI model of each system is given by

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i^{\mathrm{in}} \cup \{i\}} A_{ij} x_j + B_i u_i.$$

Notice that the inputs are not coupled. The objective for each system is to track $x_i^r = (0, 0, \Delta P_{L_i}, \Delta P_{L_i})$ and $u_i^r = \Delta P_{L_i}$, where ΔP_{L_i} denotes the local power load. In our simulations we used five systems as described in Figure 2. The local constraints for each system are as follows: $\Delta\theta_i \in [-0.1, 0.1]$ for all i, and $\Delta P_{L_1}, \Delta P_{L_5} \in [-0.5, 0.5]$, $\Delta P_{L_2}, \Delta P_{L_3} \in [-0.65, 0.65]$, and $\Delta P_{L_4} \in [-0.55, 0.55]$. Furthermore, the quadratic costs $Q_i = 4I_{s_i}$ and $R_i = I_{t_i}$ are used for all systems along the horizon. We have omitted the details on the system dynamics here. The reader is referred to [19] and the references therein for details and parameter values. We used Euler's method for discretization of the dynamics with step length of 1 *sec*. This discretization has the advantage of maintaining the sparsity patterns of the transition matrices. In all our simulation we used horizon length $N = 20$.

In Algorithm 1 the stepsizes for each system must be selected in accordance to the simple condition of Assumption 2*(iii)*. Typically, in primal-dual proximal algorithms larger stepsizes yield faster convergence. However, there is a trade-off between edge parameters $\kappa_{(i,j)}$ and node parameters, σ_i, τ_i. We selected these values empirically as follows: i) $\kappa_{(i,j)} = 10$ for all $(i,j) \in \mathcal{E}$, ii) $\sigma_i = 1$ if $d_i^{\mathrm{out}} = 1$, and $10|d_i^{\mathrm{out}} - 1|$ otherwise, iii) $\tau_i = \frac{0.99}{\max\{10 d_i^{\mathrm{out}} + \sigma_i, 10\}}$, where d_i^{out} denotes the cardinality of $\mathcal{N}_i^{\mathrm{out}}$. Notice that due to this simple local rule, removal or addition of a node only affects the neighboring nodes through d_i^{out}.

Our simulations consist of two scenarios:

Scenario 1: In the first scenario we demonstrate the *plug and play* capability of our algorithm. We consider systems $1, \ldots, 4$ with the dynamic coupling depicted in Figure 2. We assume that at time $t = 20$ system 5 is connected to systems 2 and 4. Furthermore, system 4 is disconnected from the network at time $t = 50$. Table I summarizes the load of power and network modification at given time steps. Figure 3 highlights the frequency deviation (the second state variable) for systems one and four. It is observed that the frequency control is achieved despite the load and configuration changes.

Scenario 2: In the second scenario, we considered a static

TABLE I

LOADS OF POWER AND NETWORK STRUCTURE FOR SCENARIO 1

time	5	5	20	20	35	35	50
system	1	4	2	5	5	3	4
ΔP_{L_i}	0.10	-0.12	0.08	added	0.05	-0.10	removed

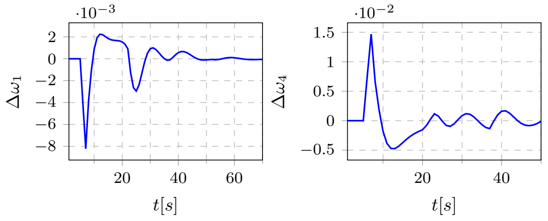

Fig. 3. Frequency deviation for systems one and four

network structure with 5 systems and load $\Delta P_{L_1} = 0.10$ with the same neighborhood structure and constraints as in the previous scenario. We compared our algorithm (referred here as PDDMPC) to [6, Algorithm 3] (DGFG) that is based on applying the fast gradient method to the dual problem. The aforementioned paper proposes solving a series of convex semidefinite program (SDP) locally at the nodes in order to select the parameters of the algorithm in a distributed fashion. In order to have a fair comparison we solved the global optimization problem using MOSEK [20]. Figure 4 demonstrates the superior performance of our scheme. The y-axis is the error defined as the norm of the difference between current primal variables and the solution in both algorithms. The x-axis denotes the total number of local iterations. Notice that DGFG requires two rounds of communication at every iteration. Furthermore, we used the randomized version of the algorithm where each system is activated independently with probability $p_i = 0.5$. It is observed that the random activation of nodes result in roughly the same number of total local iterations as the synchronous case.

VI. CONCLUSIONS

This paper introduced a fully distributed primal-dual proximal algorithm for the DMPC problem that includes both synchronous and randomized versions. In addition to simple local iterations, the stepsizes of the new algorithm are selected locally without any global coordination. Therefore, any changes to the network structure only affects the neighboring nodes. In addition, our algorithm enjoys a linear convergence rate under mild assumptions on the input and state constraints. Future works include devising efficient strategies for selecting the edge weights, and extending the algorithm for the case of lossy communications.

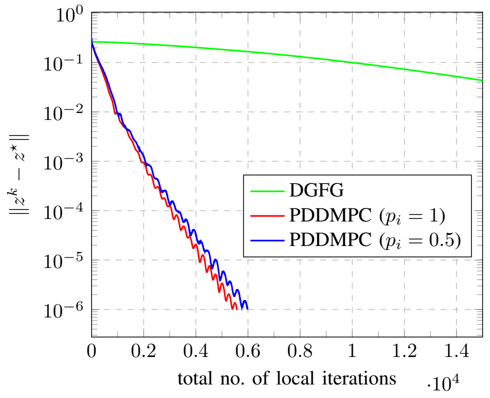

Fig. 4. Total number of local iterations: comparing synchronous PDDMPC, randomized PDDMPC and DGFG.

REFERENCES

[1] Y. Wakasa, M. Arakawa, K. Tanaka, and T. Akashi, "Decentralized model predictive control via dual decomposition," in *47th IEEE Conference on Decision and Control (CDC)*, Dec 2008, pp. 381–386.

[2] W. B. Dunbar and D. S. Caveney, "Distributed receding horizon control of vehicle platoons: Stability and string stability," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 620–633, 2012.

[3] Z. Wang and C. J. Ong, "Distributed model predictive control of linear discrete-time systems with local and global constraints," *Automatica*, vol. 81, pp. 184 – 195, 2017.

[4] P. Giselsson, M. D. Doan, T. Keviczky, B. D. Schutter, and A. Rantzer, "Accelerated gradient methods and dual decomposition in distributed model predictive control," *Automatica*, vol. 49, no. 3, pp. 829–833, 2013.

[5] A. Bemporad and D. Barcelli, "Decentralized model predictive control," in *Networked control systems*. Springer, 2010, pp. 149–178.

[6] P. Giselsson, "Improved dual decomposition for distributed model predictive control," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 1203–1209, 2014.

[7] X. Hou, Y. Xiao, J. Cai, J. Hu, and J. E. Braun, "Distributed model predictive control via proximal Jacobian ADMM for building control applications," in *American Control Conference (ACC)*, May 2017, pp. 37–43.

[8] F. Farokhi, I. Shames, and K. H. Johansson, "Distributed MPC via dual decomposition and alternating direction method of multipliers," in *Distributed model predictive control made easy*. Springer, 2014, pp. 115–131.

[9] P. Trnka, J. Pekař, and V. Havlena, "Application of distributed MPC to barcelona water distribution network," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 9139 – 9144, 2011.

[10] R. Negenborn, B. D. Schutter, and J. Hellendoorn, "Multi-agent model predictive control for transportation networks: Serial versus parallel schemes," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 3, pp. 353–366, apr 2008.

[11] P. Giselsson, "A generalized distributed accelerated gradient method for distributed model predictive control with iteration complexity bounds," in *American Control Conference (ACC)*, 2013, pp. 327–333.

[12] P. Latafat, N. M. Freris, and P. Patrinos, "A new randomized block-coordinate primal-dual proximal algorithm for distributed optimization," *arXiv preprint arXiv:1706.02882*, 2017.

[13] P. Latafat and P. Patrinos, "Asymmetric forward–backward–adjoint splitting for solving monotone inclusions involving three operators," *Computational Optimization and Applications*, vol. 68, no. 1, pp. 57–93, Sep 2017.

[14] R. T. Rockafellar, *Convex analysis*. Princeton University Press, 1997.

[15] P. Latafat, L. Stella, and P. Patrinos, "New primal-dual proximal algorithm for distributed optimization," in *55th IEEE Conference on Decision and Control (CDC)*, 2016, pp. 1959–1964.

[16] H. H. Bauschke and P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*. Springer Science & Business Media, 2011.

[17] B. O'Donoghue, G. Stathopoulos, and S. Boyd, "A splitting method for optimal control," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2432–2442, Nov 2013.

[18] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[19] S. Riverso, M. Farina, and G. Ferrari-Trecate, "Plug-and-play decentralized model predictive control for linear systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pp. 2608–2614, 2013.

[20] MOSEK ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 63).*, 2015.