A distributed framework for linear adaptive MPC

Anilkumar Parsi*, Ahmed Aboudonia*, Andrea Iannelli, John Lygeros, and Roy S. Smith

***Abstract*— Adaptive model predictive control (MPC) robustly ensures safety while reducing uncertainty during operation. In this paper, a distributed version is proposed to deal with network systems featuring multiple agents and limited communication. To solve the problem in a distributed manner, structure is imposed on the control design ingredients without sacrificing performance. Decentralized and distributed adaptation schemes that allow for a reduction of the uncertainty online compatibly with the network topology are also proposed. The algorithm ensures robust constraint satisfaction, recursive feasibility and finite gain ℓ_2 stability, and yields lower closed-loop cost compared to robust distributed MPC in simulations.**

I. Introduction

In the increasing number of applications involving complex interconnected networks, advanced control strategies are required to ensure safety while coping with coupled dynamics and constraints. A candidate is model predictive control (MPC), where a model of the system is used to optimize over state and control trajectories and guarantee constraint satisfaction [1]. Robust MPC methods take into account uncertainty in the model, but they might lead to conservative actions when the uncertainty is large. Adaptive MPC (AMPC) is a technique which ensures robustness while updating the uncertainty using online identification (e.g. see [2], [3], [4]). In AMPC, uncertainty is captured by a parameteric state space model. Uncertainty bounds are updated using set membership identification [5], while a tube MPC approach ensures robust constraint satisfaction. Two current limitations motivated this work, and are addressed by the proposed algorithms. The first is that the optimization problem which is solved online grows combinatorially with the state dimension. Although this was partially ameliorated in [6], the number of constraints still grows exponentially with the number of uncertain parameters The second is that AMPC uses a centralized optimization problem and identification scheme, which cannot be applied to interconnected networks without a central unit.

Centralized control of interconnected systems may suffer loss of privacy, lack of robustness against failure and high computational cost. Thus, distributed control has become a strong contender for these systems [7]. In particular, distributed MPC (DMPC) has gained attention due to its ability to consider constraints [8]. In many DMPC schemes, a network is decomposed into smaller subsystems. Each subsystem solves a local optimization problem iteratively while sharing information with other subsystems. This yields distributed optimal control problems that can be solved using distributed optimization (e.g. see [9]). Various efforts have been devoted to developing robust DMPC schemes for uncertain systems (e.g. see [10], [11]). However, none of the schemes consider online identification of the network's parameters.

In this work, we propose a distributed AMPC (DAMPC) algorithm suitable for control of interconnected networks. The key observation is that, for interconnected networks, the state space matrices and constraints are sparse and structured, and this is leveraged to modify the AMPC optimization problem. The optimization parameters and variables are structured such that the problem can be solved using distributed optimization, while reducing the performance loss. The number of variables and constraints in the resulting problem increases only linearly with the number of agents in the network. The MPC optimization problem is solved in a distributed fashion using alternating direction method of multipliers (ADMM) [9]. Another novel contribution of this paper is the proposal of identification schemes for an interconnected network with no central unit, which has not been studied much in literature. Specifically, two different set membership algorithms, decentralized and distributed, are proposed. In the former, the agents exchange only the state measurements with their neighbors. In the latter, the bounds of shared parameters are also exchanged with the neighboring agents, leading to better performance. The resulting MPC algorithm ensures robust constraint satisfaction, recursive feasibility and closed loop stability of the system. To the best of our knowledge, this is the first DMPC algorithm that guarantees robustness to parameteric uncertainty and performs online identification. We show the features of the proposed algorithms using an uncertain mass spring damper system.

Notation: The Minkowski sum operator is denoted by $\oplus$. The matrix O_{ij} represents a zero matrix with i rows and j columns, and $||x||_Q^2$ denotes $x^\top Qx$. The i^{th} row of a matrix A is denoted by $[A]_i$, and $\mathbb{N}_1^a$ denotes the set of integers from 1 to a.

This work has been partially supported by the Swiss National Science Foundation under grant no. 200021_178890.

The authors are with the Department of Electrical Engineering, Automatic Control Lab, ETH, Zürich 8092, Switzerland aparsi/ahmedab/iannelli/lygeros/rsmith@control.ee.ethz.ch.

*Both the authors contributed equally to this work.

II. Adaptive MPC

In this section, the robust adaptive model predictive control (AMPC) algorithm proposed in [2] is briefly described. Consider a discrete-time, linear time-invariant system with state $x_k \in \mathbb{R}^n$, control input $u_k \in \mathbb{R}^m$ and disturbance w_k at the time step k, and system dynamics

$$x_{k+1} = A(\theta)x_k + B(\theta)u_k + w_k, \tag{1}$$

where $\theta \in \mathbb{R}^p$ is a constant parameter and its true value θ^* is unknown. The state matrices are parameterized as

$$A(\theta) = A_0 + \sum_{i=1}^{p} A_i[\theta]_i, \quad B(\theta) = B_0 + \sum_{i=1}^{p} B_i[\theta]_i, \tag{2}$$

where θ belongs to the known bounded polytope $\Theta_0 := \{\theta | H_\theta \theta \leq h_{\theta_0}\}$, such that $\theta^* \in \Theta_0$ and $H_\theta \in \mathbb{R}^{n_\theta \times p}$. The disturbance w_k lies in a known bounded polytope $\mathbb{W}$. The states and inputs of the system must satisfy the constraints described by the compact polytope $\mathbb{Z} = \{(x_k, u_k) \big| Fx_k + Gu_k \leq \mathbf{1}\}$, where $F \in \mathbb{R}^{nz \times n}$ and $G \in \mathbb{R}^{nz \times m}$. The objective is to regulate the system state from the initial condition x_0 to the origin, while satisfying the constraints for all possible realizations of uncertainty θ and disturbance w_k.

Set-membership identification [5] is performed to update the parameter set, which is given by $\Theta_k := \{\theta \in \mathbb{R}^p | H_\theta \theta \leq h_{\theta_k}\}$. The bounds $h_{\theta_k} \in \mathbb{R}^{n_\theta}$ are updated online using a linear program. To ensure robustness against parameter uncertainties, a tube MPC approach is used [12]. The control input in the MPC optimization problem is parameterized as $u_{l|k} = Kx_{l|k} + v_{l|k}$, where subscript $\{\cdot\}_{l|k}$ denotes the l steps ahead prediction at time step k, $v_{l|k}$ is the MPC optimization variable and K is a control gain stabilizing all the plants in Θ_0. A state tube is defined using a sequence of sets $\{\mathbb{X}_{l|k}\}$ satisfying

$$\mathbb{X}_{0|k} \ni \{x_k\}, \quad \mathbb{X}_{l+1|k} \supseteq A(\theta)\mathbb{X}_{l|k} \oplus B(\theta)u_{l|k} \oplus \mathbb{W}, \tag{3}$$

for all $\theta \in \Theta_k$ and $l \in \mathbb{N}_0^{N-1}$, where N is the prediction horizon. The sets $\mathbb{X}_{l|k}$ are parameterized according to $\mathbb{X}_{l|k} = \{z_{l|k}\} \oplus \alpha_{l|k}\mathbb{X}_0$, where $z_{l|k} \in \mathbb{R}^n, \alpha_{l|k} \in \mathbb{R}_{\geq 0}$ are optimization variables and $\mathbb{X}_0$ is designed offline using the matrix $H_x \in \mathbb{R}^{n_x \times n}$ as

$$\mathbb{X}_0 := \{x | H_x x \leq \mathbf{1}\} = \mathrm{co}\{x^1, x^2, \dots, x^q\}, \tag{4}$$

where $x^i, i \in \mathbb{N}_1^q$ are q prespecified vertices and co$\{.\}$ denotes their convex hull. The states in the set $\mathbb{X}_{N|k}$ are constrained to lie in a terminal set $\mathbb{X}_T := \{x | H_x x \leq \bar{\alpha}\mathbf{1}\}$, that is robustly invariant under the terminal controller $u = Kx$. The constant $\bar{\alpha}$ is computed offline as the maximum value of α for which $\alpha\mathbb{X}_0 \in \mathbb{Z}$.

In the MPC problem, the state and input constraints, set inclusions for the tube, and the terminal constraints are formulated in terms of the optimization variables $\{z_{l|k}, v_{l|k}, \alpha_{l|k}\}$. For simplicity, we define the notation

$$\begin{aligned} x^j_{l|k} &= z_{l|k} + \alpha_{l|k}x^j, & d^j_{l|k} &= A_0 x^j_{l|k} + B_0 u^j_{l|k} - z_{l+1|k}, \\ u^j_{l|k} &= Kx^j_{l|k} + v_{l|k}, & D^j_{l|k} &= D(x^j_{l|k}, u^j_{l|k}), \end{aligned} \tag{5}$$

where $j \in \mathbb{N}_1^q, l \in \mathbb{N}_0^{N-1}$, and the matrix $D(x_k, u_k) = [A_1 x_k + B_1 u_k \ \dots \ A_p x_k + B_p u_k]$. The vectors $\bar{f}, \bar{w}$ are precomputed offline such that $[\bar{f}]_i = \max_{x \in \mathbb{X}_0} [F + GK]_i x$, $i \in \mathbb{N}_1^{nz}$ and $[\bar{w}]_j = \max_{w \in \mathbb{W}} [H_x]_j w$, $j \in \mathbb{N}_1^{n_x}$. A quadratic cost function is used in MPC following a certainty-equivalence approach. For this purpose, an estimate of the parameter $(\hat{\theta}_k)$ is obtained using a least mean squares filter [13] and projected onto the set Θ_k at each time step. The final MPC optimization problem can be written as

$$\min_{\substack{z_{l|k}, v_{l|k}, \\ \alpha_{l|k}, \Lambda^j_{l|k}}} \sum_{l=0}^{N-1} \left(||\hat{x}_{l|k}||^2_Q + ||\hat{u}_{l|k}||^2_R \right) + ||\hat{x}_{N|k}||^2_P, \tag{6a}$$

$$\text{s.t.} \qquad -H_x z_{0|k} - \alpha_{0|k}\mathbf{1} \leq -H_x x_k, \tag{6b}$$

$$(F + GK)z_{l|k} + Gv_{l|k} + \alpha_{l|k}\bar{f} \leq \mathbf{1}, \tag{6c}$$

$$\Lambda^j_{l|k} h_{\theta_k} + H_x d^j_{l|k} - \alpha_{l+1|k}\mathbf{1} \leq -\bar{w}, \tag{6d}$$

$$H_x D^j_{l|k} = \Lambda^j_{l|k} H_\theta, \tag{6e}$$

$$\hat{x}_{0|k} = x_k, \quad A(\hat{\theta}_k)\hat{x}_{l|k} + B(\hat{\theta}_k)\hat{u}_{l|k} = \hat{x}_{l+1|k}, \tag{6f}$$

$$z_{N|k} = 0, \quad \alpha_{N|k} \leq \bar{\alpha}, \tag{6g}$$

where $j \in \mathbb{N}_1^q$, $l \in \mathbb{N}_0^{N-1}$ and P, Q, R are positive definite matrices. The state tube is initialized in (6b) and the state and input constraints are reformulated in terms of the optimization variables in (6c). The state tube dynamics are propagated inside the prediction horizon in (6d) and (6e) using the Lagrange multiplier matrices $\Lambda^j_{l|k} \in \mathbb{R}^{n_x \times n_\theta}$, $j \in \mathbb{N}_1^q, l \in \mathbb{N}_0^{N-1}$, which have non-negative entries. The certainty equivalence trajectory is defined in (6f) and the terminal constraints are shown in (6g). Note that the number of variables and constraints in (6) depends on q, i.e., the number of vertices in $\mathbb{X}_0$ (4), which increases combinatorially with the state dimension.

III. Distributed adaptive MPC

We present here a DAMPC algorithm which enables a network of agents to be controlled using an adaptive MPC approach. The algorithm requires no central unit for online implementation. Moreover, the size of the MPC optimization problem only linearly grows with the number of agents in the network.

Consider an interconnected system of S agents (also referred to as subsystems). Each agent has uncertain parameters, some of which might be common to multiple agents. The goal is to impose structure in the design matrices and sets defined in Section II so that (6) can be solved in a distributed manner. As a preamble, note that variables and matrices corresponding to agent s are denoted using the subscript, for example $A_{0,s}$, $v_{l|k,s}$. Then, agent s in the interconnected system obeys

$$x_{k+1,s} = A_{\mathrm{d},s}(\theta_s)x_{k,\mathcal{N}_s} + B_{\mathrm{d},s}(\theta_s)u_{k,s} + w_{k,s}, \tag{7}$$

where the subscript d is used to differentiate the matrices $A_{\mathrm{d},s}$ from A_i in (2). The set of neighbors $\mathcal{N}_s$ contains all the agents coupled dynamically or through the constraints with agent s, including itself. The vector $x_{k,\mathcal{N}_s}$

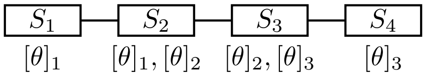

Fig. 1: Example interconnected network. The parameters affecting each agent are reported below.

comprises states of all the agents in the set $\mathcal{N}_s$, and $\theta_s \in \mathbb{R}^{p_s}$ consists of parameters affecting agent s. It is assumed that agents can communicate with all of their neighbors, and that their inputs are decoupled.

For the sake of clarity, a notional network topology is used to illustrate the structure imposed on the MPC problem. Consider the 4-agent network shown in Figure 1, with each agent having three states and two inputs. For this example, the state matrix has the structure

$$A(\theta) = \begin{bmatrix} \boxed{A_{\mathrm{d},1}(\theta_1)} & O_{33} & O_{33} \\ \multicolumn{2}{c}{\boxed{A_{\mathrm{d},2}(\theta_2)}} & O_{33} \\ O_{33} & \multicolumn{2}{c}{\boxed{A_{\mathrm{d},3}(\theta_3)}} \\ O_{33} & O_{33} & \boxed{A_{\mathrm{d},4}(\theta_4)} \end{bmatrix},$$

while the input matrix $B(\theta)$ has a block structure connecting each agent's inputs to its own states. We assume that the constraints to be satisfied by the agent s are given by the compact polytope $\mathbb{Z}_s = \left\{(x_{k,\mathcal{N}_s}, u_{k,s}) \middle| F_s x_{k,\mathcal{N}_s} + G_s u_{k,s} \leq \mathbf{1}\right\}$. Recall that states of non-neighboring agents are not coupled by constraints, and the input constraints are fully decoupled.

A. Structured design matrices

The control gain K is designed such that the feedback can be implemented using a distributed feedback controller. For this purpose, in each row of K corresponding to agent s, only the columns corresponding to the set $\mathcal{N}_s$ can be nonzero. These values can be extracted to obtain the control gain K_s for each agent, and the MPC control input satisfies $u_{l|k,s} = K_s x_{l|k,\mathcal{N}_s} + v_{l|k,s}$. The structured gain K can be computed offline, for example as suggested in [14], [15]. Although [14] is tailored for centralized MPC, this method can be extended to distributed systems by imposing appropriate structure on the decision variables. Furthermore, although [15] is proposed for nominal MPC, it can be extended to the uncertain system (1) by ensuring that the corresponding constraints are satisfied for all vertices of Θ_0. For the example in Figure 1, the structure of K is

$$K = \begin{bmatrix} \boxed{K_1} & O_{23} & O_{23} \\ \multicolumn{2}{c}{\boxed{K_2}} & O_{23} \\ O_{23} & \multicolumn{2}{c}{\boxed{K_3}} \\ O_{23} & O_{23} & \boxed{K_4} \end{bmatrix}. \tag{8}$$

The matrix H_x defined in (4) is structured so that, in each row, only the columns corresponding to a single agent can be nonzero. This results in a block matrix structure, where each block represents the structure of the state tube of the corresponding agent as

$$\mathbb{X}_{0,s} := \{x_s | H_{x,s} x_s \leq \mathbf{1}\} = \mathrm{co}\{x_s^1, x_s^2, \ldots, x_s^{q_s}\}. \tag{9}$$

The structure (9) implies that the number of vertices to be evaluated depends on $\sum_{s=1}^{S} q_s$ instead of q, resulting in linear growth in the number of variables and constraints with the number of agents. For the example shown in Figure 1, the structure of H_x is

$$H_x = \begin{bmatrix} \boxed{H_{x,1}} & O_{83} & O_{83} & O_{83} \\ O_{83} & \boxed{H_{x,2}} & O_{83} & O_{83} \\ O_{83} & O_{83} & \boxed{H_{x,3}} & O_{83} \\ O_{83} & O_{83} & O_{83} & \boxed{H_{x,4}} \end{bmatrix}, \tag{10}$$

where each agent is using 8 constraints to define $X_{0,s}$. The computation of a structured H_x while ensuring robust invariance of the terminal set is a difficult task, and is the topic of future research. The block structure (9) enables an additional degree of freedom in parameterizing the state tube according to $\mathbb{X}_{l|k,s} = z_{l|k,s} \oplus \alpha_{l|k,s} \mathbb{X}_{0,s}$, where $\alpha_{l|k,s}$ chosen by each subsystem need not be equal.

The matrix H_θ is structured so that different parameters can have a nonzero coefficient in the same hyperplane only if they affect a common agent. The structure chosen in H_θ affects the Lagrange multipliers $\Lambda_{l|k}^j$ used to build the state tube. If a constraint affects multiple agents, they must use the same Lagrange multipliers, resulting in a large number of shared variables. To solve this problem, redundant constraints are added to $\{H_\theta, h_{\theta_k}\}$, and denoted as $\{\tilde{H}_\theta, \tilde{h}_{\theta_k}\}$. These constraints are formed by repeating constraints affecting multiple agents by the number of agents affected. The resulting matrix $\tilde{H}_\theta$ has a block representation, where each block $\tilde{H}_{\theta,s}$ and the corresponding terms $\tilde{h}_{\theta_k,s}$ represent all the constraints on the parameters affecting agent s. For the example in Figure 1, let the matrix H_θ be such that the constraints define upper and lower bounds on the parameters $[\theta]_1, [\theta]_2, [\theta]_3$, and two constraints jointly on the parameters $[\theta]_1, [\theta]_2$. This results in the structure

$$H_\theta = \begin{bmatrix} \boxed{H_{\theta\text{-}1}} & O_{21} & O_{21} \\ \multicolumn{2}{c}{\boxed{H_{\theta\text{-}1,2}}} & O_{21} \\ O_{21} & \boxed{H_{\theta\text{-}2}} & O_{21} \\ O_{21} & O_{21} & \boxed{H_{\theta\text{-}3}} \end{bmatrix},\ \tilde{H}_\theta = \begin{bmatrix} \boxed{\tilde{H}_{\theta,1}} & O_{21} & O_{21} \\ \multicolumn{2}{c}{\boxed{\tilde{H}_{\theta,2}}} & O_{61} \\ O_{41} & \multicolumn{2}{c}{\boxed{\tilde{H}_{\theta,3}}} \\ O_{21} & O_{21} & \boxed{\tilde{H}_{\theta,4}} \end{bmatrix},$$

where the subscript i, j in $H_{\theta\text{-}i,j}$ indicate the parameters involved in the constraints defined by the block.

Due to the additional rows in $\tilde{H}_\theta$, additional columns are added to $\Lambda_{l|k}^j$ and the new matrix is denoted as $\tilde{\Lambda}_{l|k}^j$. In $\tilde{\Lambda}_{l|k}^j$, each row corresponding to $H_{x,s}$ has nonzero elements only in the columns corresponding to $H_{\theta,s}$. For the example system, this results in the structure

$$\tilde{\Lambda}_{l|k}^j = \begin{bmatrix} \boxed{\Lambda_{l|k,1}^j} & O_{86} & O_{84} & O_{82} \\ O_{82} & \boxed{\Lambda_{l|k,2}^j} & O_{84} & O_{82} \\ O_{82} & O_{86} & \boxed{\Lambda_{l|k,3}^j} & O_{82} \\ O_{82} & O_{86} & O_{84} & \boxed{\Lambda_{l|k,4}^j} \end{bmatrix}. \tag{11}$$

Finally, the cost matrices P, Q, and R have a block diagonal structure, where the corresponding blocks for each agent are denoted as P_s, Q_s, and R_s.

Algorithm 1 Distributed identification

```
Perform decentralized identification (12)-(13)
Communicate $\theta^{\mathrm{ub}}_{k,s}, \theta^{\mathrm{lb}}_{k,s}$ with corresponding neighbors
Compute the bounds according to (14)
Update $h_{\theta_k,s}$ using $\bar{\theta}_{k,s}$ and $\underline{\theta}_{k,s}$.
```

B. Set membership identification

To perform DAMPC, each agent defines the feasible parameter set $\Theta_{k,s} := \{\theta_s | \tilde{H}_{\theta,s}\theta_s \leq \tilde{h}_{\theta_k,s}\}$ which is updated using set membership identification. However, the identification method proposed in [2] cannot be used here, since it requires a central agent with access to all the measurements. We propose two approaches to update $\Theta_{k,s}$ with local knowledge. At each time step, the state measurements are shared with the neighbors, and used to construct a set of non-falsified parameters

$$\begin{aligned}\Delta_{k,s} :=& \left\{ \theta_s \;\middle|\; x_{k+1,s} - A_{\mathrm{d},s}(\theta_s)x_{k,\mathcal{N}_s} - B_{\mathrm{d},s}(\theta_s)u_{k,s} \in \mathbb{W}_s, \right\} \\ =& \left\{ \theta_s \mid H_{\Delta,s}\theta_s \leq h_{\Delta,s} \right\}. \end{aligned} \tag{12}$$

The parameter set is then updated by computing $[\tilde{h}_{\theta_k,s}]_i$ as the optimal value of

$$\max_{\theta_s} [\tilde{H}_{\theta,s}]_i\theta_s, \quad \text{s. t. } \begin{bmatrix} \tilde{H}_{\theta,s} \\ H_{\Delta,s} \end{bmatrix} \theta_s \leq \begin{bmatrix} \tilde{h}_{\theta_{k-1},s} \\ h_{\Delta,s} \end{bmatrix}, \tag{13}$$

for $i \in \mathbb{N}_1^{n_\theta,s}$. Problem (12)-(13) provides a decentralized identification scheme, where agent s updates its own parameters using the measurements of its neighbors. An alternative is distributed identification, where parameter bounds are also communicated with neighbors. This can be useful when multiple agents are affected by the same parameter uncertainty, but have different identification performance. Algorithm 1 details the procedure for such a distributed identification scheme. As a prerequisite, the algorithm requires that the polytope $\Theta_{k,s}$ has hyperplanes orthogonal to each parameter's axis. The corresponding elements in $h_{\theta_k,s}$ are the upper ($[\theta^{\mathrm{ub}}_{k,s}]_i$) and lower ($[\theta^{\mathrm{lb}}_{k,s}]_i$) bounds of the i^{th} parameter, as computed by agent s. In the first step of the algorithm, decentralized identification is performed to obtain $h_{\theta_k,s}$. Then, the bounds of the shared parameters are exchanged with the corresponding neighbors. The upper and lower bounds are then updated using all the information received as

$$[\bar{\theta}_{k,s}]_i = \min_{j\in\mathcal{N}_s}([\theta^{\mathrm{ub}}_{k,j}]_{\hat{i}}), \quad [\underline{\theta}_{k,s}]_i = \max_{j\in\mathcal{N}_s}([\theta^{\mathrm{lb}}_{k,j}]_{\hat{i}}), \tag{14}$$

where $[\theta^{\mathrm{ub}}_{k,j}]_{\hat{i}}, [\theta^{\mathrm{lb}}_{k,j}]_{\hat{i}}$ denote the upper and lower bounds of the parameter $[\theta_{k,s}]_i$ as computed by the agent j. The values of $h_{\theta_k,s}$ are updated using $[\bar{\theta}_{k,s}]_i, [\underline{\theta}_{k,s}]_i$.

C. Distributed optimization

Incorporating the structure described in Section III-A, the optimization problem (6) can be reformulated as

$$\begin{aligned} \min_{\substack{z_{l|k,s}, v_{l|k,s} \\ \alpha_{l|k,s}, \tilde{\Lambda}^j_{l|k,s}}} \sum_{s=1}^{S}\sum_{l=0}^{N-1} & \left(||\hat{x}_{l|k,s}||^2_{Q_s} + ||\hat{u}_{l|k,s}||^2_{R_s}\right) + ||\hat{x}_{N|k,s}||^2_{P_s} \\ \text{s.t.} \qquad & -H_{x,s}z_{0|k,s} - \alpha_{0|k,s}\mathbf{1} \leq -H_{x,s}x_{k,s}, \\ & F_{\mathrm{cl},s}z_{l|k,s} + G_s v_{l|k,s} + \alpha_{l|k,s}\bar{f}_s \leq \mathbf{1}, \\ & \tilde{\Lambda}^j_{l|k,s}\tilde{h}_{\theta_k,s} + H_{x,s}d^j_{l|k,s} - \alpha_{l+1|k,s}\mathbf{1} \leq -\bar{w}_s, \\ & H_{x,s}D^j_{l|k,s} = \tilde{\Lambda}^j_{l|k,s}\tilde{H}_{\theta,s}, \\ & A_s(\hat{\theta}_{k,s})\hat{x}_{l|k,\mathcal{N}_s} + B_s(\hat{\theta}_{k,s})\hat{u}_{l|k,s} = \hat{x}_{l+1|k,s}, \\ & \hat{x}_{0|k,s} = x_{k,s}, \quad z_{N|k,s} = 0, \quad \alpha_{N|k,s} \leq \bar{\alpha}_s, \end{aligned} \tag{15}$$

where $F_{\mathrm{cl},s} = F_s + G_s K_s$, $j \in \mathbb{N}_1^{v_s}$, $l \in \mathbb{N}_0^{N-1}$, $s \in \mathbb{N}_1^S$, and the constants $\bar{f}_s, \bar{w}_s, \bar{\alpha}_s$ are calculated offline. The parameter estimate $\hat{\theta}_{k,s}$ is computed by agent s using a least mean squares filter. Note that the constraints in (15) are the same as those in (6), but with a distributed structure.

Proposition 1: The DAMPC algorithm guarantees robust constraint satisfaction and recursive feasibility of the optimization problem. In addition, the closed loop defined by the system (1) and the DAMPC scheme is finite gain ℓ_2 stable.

We provide only an outline of the proof that follows similar lines to those of Lemma 5 and Theorem 14 from [2]. In addition to the structure imposed on K and H_x, which does not affect the proof, the proposed changes to the AMPC algorithm are: a distributed identification scheme; repeated constraints in the parameter set; and additional Lagrange multipliers. The added constraints and variables are redundant, and do not affect the feasibility of the optimization problem. The proposed decentralized and distributed identification algorithms guarantee by design that $\theta^* \in \Theta_k$ and $\Theta_k \subseteq \Theta_{k-1}$, and thus, recursive feasibility is preserved.

To ensure finite gain ℓ_2 stability, there must exist constants $c_0, c_1, c_2 \geq 0$ such that for $m \in \mathbb{N}_+$

$$\begin{aligned} \sum_{s=0}^{S}\sum_{k=0}^{m} ||x_{k,s}||^2 \leq & \sum_{s=0}^{S}\big(c_0||x_{0,s}||^2 \\ & + c_1||\hat{\theta}_{0,s} - \theta^*||^2 + c_2\sum_{k=0}^{m}||w_{k,s}||^2\big). \end{aligned} \tag{16}$$

Denoting the optimal value of the cost function in (15) as V_k, the change in its value can be bounded as

$$V_{k+1} - V_k \leq \sum_{s=0}^{S}\left(-c||x_{k,s}||^2 + c_A||x_{k+1,s} - \hat{x}_{1|k,s}||^2\right), \tag{17}$$

using $c, c_A \geq 0$. The upper bound in (17) is computed by constructing a candidate trajectory at time step $k+1$, using the parameter estimates $\hat{\theta}_{k+1,s}$ and shifting the optimal solution $v^*_{l|k,s}$ by one time step. The last control input variable $v_{N|k+1,s}$ is set to 0. The second term in the upper bound is the one step ahead prediction error. The least mean squares filters guarantee that the sum of squared prediction error can always be bounded using the initial error in the parameter estimate and the sum of the squared disturbances. This has been proven in [2] for a centralized filter, and the bound is also valid for the DAMPC algorithm although it uses multiple copies of some parameters and a distributed filter. It can also

TABLE I: ADMM variables computed by the second agent in the network example in Fig. 1

Step 1	Step 2	Step 3
$C_{\mathcal{N}_2} = \left[C_{\mathcal{N}_2,1}^\top \; C_{\mathcal{N}_2,2}^\top \; C_{\mathcal{N}_3,2}\right]^\top$ $C_{\mathcal{N}_2,s} = \left[z_{l\|k,s}^{(2)\top}, \alpha_{l\|k,s}^{(2)\top}, \hat{x}_{l\|k,s}^{(2)\top}\right]^\top_{\left\{\begin{smallmatrix} s \in \mathbb{N}_1^3, \\ l \in \mathbb{N}_0^{N-1}\end{smallmatrix}\right\}}$ $E_2 = \left[v_{l\|k,2}^{(2)\top}, \mathrm{vec}(\tilde{\Lambda})_{l\|k,2}^{(2)\top}, \hat{u}_{l\|k,2}^{(2)\top}\right]^\top_{\{l \in \mathbb{N}_0^{N-1}\}}$	T_2	$Y_{\mathcal{N}_2}$

be shown that V_k can be upper bounded in the feasible set by a quadratic function of the states $x_{k,s}$. This means that summing (17) over k results in (16).

We now cast the distributed optimal control problem (15) into the general-form consensus problem in [9, Section 7.2] so that it can be solved in a distributed fashion using ADMM. Note that two neighbours s_1 and s_2 share the decision variables $z_{l|k,s_1}$, $z_{l|k,s_2}$, $\alpha_{l|k,s_1}$, $\alpha_{l|k,s_2}$, $\hat{x}_{l|k,s_1}$ and $\hat{x}_{l|k,s_2}$. Hence, we define $C_{\mathcal{N}_s}$ to be a concatenated vector comprising local copies of all decision variables that subsystem s shares with its neighbours (i.e. $C_{\mathcal{N}_s} = \left[z_{l|k,\sigma}^{(s)\top}, \alpha_{l|k,\sigma}^{(s)\top}, \hat{x}_{l|k,\sigma}^{(s)\top}\right]^\top_{\{l \in \mathbb{N}_0^{N-1}, \sigma \in \mathcal{N}_s\}}$ where the superscript $(\cdot)^{(s)}$ denotes the local copy computed by subsystem s). We also define $Y_{\mathcal{N}_s}$ as the vector of dual variables whose size is the same as that of $C_{\mathcal{N}_s}$. For any $\sigma \in \mathcal{N}_s$, we define the vector $C_{\mathcal{N}_s,\sigma} = \left[z_{l|k,\sigma}^{(s)\top}, \alpha_{l|k,\sigma}^{(s)\top}, \hat{x}_{l|k,\sigma}^{(s)\top}\right]^\top_{\{l \in \mathbb{N}_0^{N-1}\}}$ which is a subvector in $C_{\mathcal{N}_s}$. On the other hand, a subsystem s does not share the decision variables $v_{l|k,s}$, $\tilde{\Lambda}_{l|k,s}$ and $\hat{u}_{l|k,s}$. Thus, we define E_s to be a concatenated vector comprising all decision variables that are not shared by subsystem s (i.e. $E_s = \left[v_{l|k,s}^{(s)\top}, \mathrm{vec}(\tilde{\Lambda})_{l|k,s}^{(s)\top}, \hat{u}_{l|k,s}^{(s)\top}\right]^\top_{\{l \in \mathbb{N}_0^{N-1}\}}$ where $\mathrm{vec}(\cdot)$ is the matrix vectorization operator). Finally, we define T as a concatenated vector comprising a global copy of all shared decision variables in the network. We denote the subvectors of T corresponding to $C_{\mathcal{N}_s}$ and $C_{\mathcal{N}_s,s}$ by $T_{\mathcal{N}_s}$ and T_s, respectively. Note that for any $s \in \mathbb{N}_1^S$ and $\sigma \in \mathcal{N}_s$, the vectors $C_{\mathcal{N}_s,s}$ and $C_{\mathcal{N}_\sigma,s}$ correspond to the same subvector in T, that is, T_s.

By defining the local function $f_s(E_s, C_{\mathcal{N}_s})$ which encodes the local objective function and constraints of subsystem s through indicator functions, the general-form consensus problem is given by

$$\min_{E_s, C_{\mathcal{N}_s}} \sum_{s=1}^{S} f_s(E_s, C_{\mathcal{N}_s}) \; s.t. \; C_{\mathcal{N}_s} = T_{\mathcal{N}_s} \; \forall s \in \mathbb{N}_1^S. \quad (18)$$

According to [9], the ADMM steps performed by subsystem s in iteration t are given by

$$\{E_s^{t+1}, C_{\mathcal{N}_s}^{t+1}\} = \underset{E_s, C_{\mathcal{N}_s}}{\arg\min} \begin{pmatrix} f_s(E_s, C_{\mathcal{N}_s}) + Y_{\mathcal{N}_s}^{t\top} C_{\mathcal{N}_s} \\ + \frac{\rho}{2} \left\| C_{\mathcal{N}_s} - T_{\mathcal{N}_s} \right\|_2 \end{pmatrix},$$

$$T_s^{t+1} = \frac{1}{|\mathcal{N}_s|} \sum_{\sigma \in \mathcal{N}_s} C_{\mathcal{N}_\sigma,s}^{t+1}, \quad (19)$$

$$Y_{\mathcal{N}_s}^{t+1} = Y_{\mathcal{N}_s}^{t} + \rho(C_{\mathcal{N}_s}^{t+1} - T_{\mathcal{N}_s}^{t+1}),$$

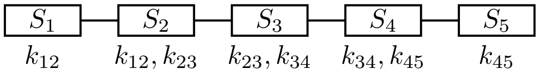

Fig. 2: Mass-spring-damper interconnected system.

where ρ is a tuning scalar and $|\cdot|$ is the set cardinality operator. The ADMM algorithm (19) updates iteratively, for all $s \in \mathcal{N}_1^S$, the primal variables $C_{\mathcal{N}_s}$ and E_s, the global copies of all shared decision variables T_s, and the dual variables $Y_{\mathcal{N}_s}$, in a sequential manner. Although the vector T is a global copy of the shared variables, the update of its subvectors T_s in the second ADMM step depends only on local information given by subsystem s and its neighbours. We also note that a few parameters, e.g. ρ and N, should be known by all subsystems, but these are constant and hence can be communicated in the offline phase as in [16]. For the sake of clarity, Table I shows the decision variables computed by the second agent in each ADMM step for the network in Fig. 1.

IV. Illustrative example

We consider an interconnected mass-spring-damper system, consisting of five masses, connected with four springs and four dampers, as shown in Figure 2. All masses are equal to 1, and the damping coefficients are $\{2, 2, 2, 1.5\}$. The true spring constants are unknown, but lie within a bounded set. The upper and lower bounds of each spring constant are given as $\{2.4, 3.6, 2.5, 2.7\} \pm \kappa \{0.7, 1, 1, 0.7\}$, where κ is a model uncertainty scaling parameter. Each mass can be manipulated by a force, which is the control input. The dynamics are discretized using Euler discretization with a sampling time of 0.1s, with the position and velocity of each mass as the states of the system. The code used for the analyses of this example can be found in the repository[1]. The system starts from the initial state $[2, 1, 2, -1, 0, 0, 2, -1, 2, 1]^\top$. All states and inputs must lie within the bounds $[-5, 5]$, and w_k is bounded by ± 0.05. The MPC cost is defined by matrices $Q{=}I_{10}, R{=}5I_5, P{=}100I_{10}$, where I_n is an $n{\times}n$ identity matrix. The DAMPC scheme is designed with an horizon $N{=}5$. The gain matrix K is obtained by imposing structure in the procedure described in [14]. This results in a decentralized structure, where the control input of an agent is a function of only its states. The state tube matrix H_x is constructed by circumscribing the stabilized ellipsoidal region in the subspace defined by each agent's states using a polytope. The matrix H_θ is defined by upper and lower bounds on each parameter. The ADMM algorithm is implemented with $\rho{=}25$ for 400 iterations.

The closed loop performance of the DAMPC with decentralized and distributed identification schemes is compared to a distributed robust MPC (DRMPC) controller in Table II. The simulations are performed for increasing values of κ, and the costs are averaged over 25 realizations of the disturbance w_k. The DRMPC scheme

[1] https://gitlab.ethz.ch/aparsi/distributedampcpublic

TABLE II: Average reduction in closed loop cost achieved by adaptation (with both distributed and decentralized identification) for different sizes of model uncertainty.

κ	0.3	0.5	0.6	0.7	0.8	0.9	1.0
DRMPC Cost	376	417	462	527	610	730	1003
% decrease decentralized	0.0	1.2	2.7	4.0	5.6	6.8	9.5
% decrease distributed	0.1	2.1	4.4	6.6	9.3	11.3	15.1

is obtained by using the same design matrices as in DAMPC but not performing adaptation. The reduction in the cost achieved by using the two proposed identification schemes is also shown in the last two rows. For low values of κ, the performance of the controllers is almost identical. As κ increases, the closed loop cost is reduced when DAMPC schemes are used. The relative improvement in performance is higher when the model uncertainity is large. Moreover, distributed identification yields markedly larger cost reductions. To provide more insights on this, Figure 3 compares the performance of the proposed identification schemes. In decentralized identification, parameter bounds are updated separately by each agent. This leads to conservatism, especially in agents affected by multiple uncertainties. For example, agent 1 identifies k_{12} better than agent 2, as it is only affected by one uncertainty. The updated bounds are not shared with agent 2, resulting in larger uncertainty for $k_{12,2}$. This problem is mitigated using distributed identification, where parameter bounds are also shared. Note that the performance of distributed identification is better than the intersection of bounds obtained from decentralized identification (e.g. see k_{23} in Figure 3).

V. Conclusions

A distributed version of the AMPC algorithm is presented for interconnected systems with parametric uncertainty. The proposed algorithm is obtained by imposing structure on design matrices, such that the optimization problem can be solved in a distributed manner. The size of the resulting distributed optimization problem grows linearly with the number of agents in the network. Distributed and decentralized identification algorithms are proposed such that no centralized computations are required. The former, by leveraging communication between agents, results in tighter parameter bounds and lower closed loop costs. It is anticipated that applications such as power networks [17] and DC microgrids [18], where the coupling may depend on uncertain shared parameters, could benefit significantly of this novel scheme. The practical demonstration of these advantages is left for future work.

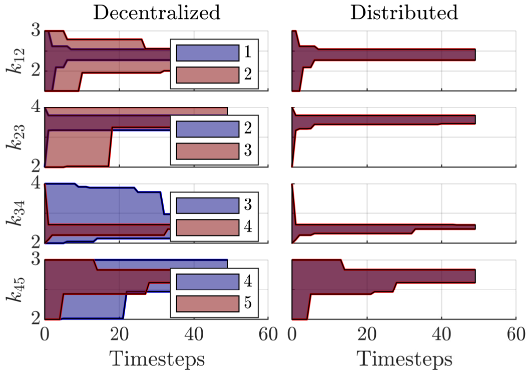

Fig. 3: Parameter bounds obtained from decentralized and distributed identification techniques. On the left, estimates of each agent are indicated in the legend. On the right, shared estimates are shown in purple.

References

[1] J. B. Rawlings and D. Q. Mayne, *Model predictive control: Theory and design.* Nob Hill Pub., 2009.

[2] M. Lorenzen, M. Cannon, and F. Allgöwer, "Robust MPC with recursive model update," *Automatica*, vol. 103, 2019.

[3] A. Parsi, A. Iannelli, and R. S. Smith, "Active exploration in adaptive model predictive control," in *59th IEEE Conference on Decision and Control (CDC)*, 2020.

[4] M. Bujarbaruah, X. Zhang, M. Tanaskovic, and F. Borrelli, "Adaptive stochastic MPC under time varying uncertainty," *IEEE Transactions on Automatic Control*, 2020.

[5] M. Milanese and A. Vicino, "Optimal estimation theory for dynamic systems with set membership uncertainty: an overview," *Automatica*, vol. 27, no. 6, pp. 997–1009, 1991.

[6] J. Köhler, E. Andina, R. Soloperto, M. A. Müller, and F. Allgöwer, "Linear robust adaptive model predictive control: Computational complexity and conservatism," in *IEEE 58th Conference on Decision and Control (CDC)*, 2019.

[7] G. Antonelli, "Interconnected dynamic systems: An overview on distributed control," *IEEE Control Systems Magazine*, vol. 33, no. 1, pp. 76–88, 2013.

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

[9] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[10] R. A. Shalmani, M. Rahmani, and N. Bigdeli, "Nash-based robust distributed model predictive control for large-scale systems," *Journal of Process Control*, vol. 88, pp. 43–53, 2020.

[11] A. Ma, K. Liu, Q. Zhang, and Y. Xia, "Distributed MPC for linear discrete-time systems with disturbances and coupled states," *Systems & Control Letters*, vol. 135, p. 104578, 2020.

[12] W. Langson, I. Chryssochoos, S. Raković, and D. Q. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, 2004.

[13] B. Hassibi, A. H. Sayed, and T. Kailath, "LMS is H_∞ optimal," in *Adaptive Control, Filtering, and Signal Processing.* Springer, 1995, pp. 65–88.

[14] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.

[15] C. Conte, 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, 2016.

[16] A. Aboudonia, A. Eichler, and J. Lygeros, "Distributed model predictive control with asymmetric adaptive terminal sets for the regulation of large-scale systems," in *21st IFAC World Congress*, 2020.

[17] 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.

[18] M. Tucci, S. Riverso, and G. Ferrari-Trecate, "Line-independent plug-and-play controllers for voltage stabilization in dc microgrids," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 3, pp. 1115–1123, 2017.