A Jacobi decomposition algorithm for distributed convex optimization in Distributed Model Predictive Control

Dang Doan, Minh; Diehl, Moritz; Keviczky, Tamas; De Schutter, Bart

DOI
10.1016/j.ifacol.2017.08.744

Publication date
2017

Document Version
Final published version

Published in
IFAC-PapersOnLine

Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 50-1 (2017) 4905–4911

A Jacobi Decomposition Algorithm for Distributed Convex Optimization in Distributed Model Predictive Control [1]

Minh Dang Doan * **Moritz Diehl** * **Tamas Keviczky** ** **Bart De Schutter** **

* *Systems Control and Optimization Laboratory, Department of Microsystems Engineering and Department of Mathematics, University of Freiburg, Germany, (e-mail: minhdang@doan.vn, moritz.diehl@imtek.uni-freiburg.de)*

** *Delft Center for Systems and Control, Delft University of Technology, The Netherlands, (e-mail: t.keviczky@tudelft.nl, b.deschutter@tudelft.nl)*

Abstract

In this paper we introduce an iterative distributed Jacobi algorithm for solving convex optimization problems, which is motivated by distributed model predictive control (MPC) for linear time-invariant systems. Starting from a given feasible initial guess, the algorithm iteratively improves the value of the cost function with guaranteed feasible solutions at every iteration step, and is thus suitable for MPC applications in which hard constraints are important. The proposed iterative approach involves solving local optimization problems consisting of only few subsystems, depending on the flexible choice of decomposition and the sparsity structure of the couplings. This makes our approach more applicable to situations where the number of subsystems is large, the coupling is sparse, and local communication is available. We also provide a method for checking a posteriori centralized optimality of the converging solution, using comparison between Lagrange multipliers of the local problems. Furthermore, a theoretical result on convergence to optimality for a particular distributed setting is also provided.

Keywords: distributed optimization, Jacobi algorithm, distributed model predictive control, cooperative optimization algorithm

1. INTRODUCTION

Distributed model predictive control (DMPC) is an approach to bring MPC into applications with large-scale systems, for example in multi-agent infrastructure networks or process control. Since its first major introduction in Jia and Krogh (2001); Camponogara et al. (2002) there has been active research in this field that has lead to a variety of branches that differ in the structure of communications, cooperative or non-cooperative approach, level of coordination, amount of information exchange, iterative or non-iterative approach, etc. (Scattolini, 2009).

We focus on a direction of DMPC that is based on the use of convex optimization and primal decomposition of the centralized problem, which has a nice property to always yield a feasible solution, even when the iteration does not have enough time to converge. Since the capability to guarantee feasibility is critical to MPC, this property is very important. Initial work in this direction were carried out by Venkat in Venkat et al. (2005, 2008), when he proposed a DMPC scheme for dynamically coupled systems called *feasible-cooperation MPC* (FC-MPC), based on a parallel synchronous approach for cooperative optimization. This scheme works only for input-coupled linear time-invariant subsystem dynamics without state constraints, and is not applicable to problems with constraints between subsystems. Later, Stewart's work (Stewart et al., 2010, 2011) extended Venkat's scheme for the case of state constraints. However this scheme can only achieve suboptimality when coupled constraints are present. These results were also documented in the book Rawlings and Mayne (2009). Further developments on determining optimal weights of convex combinations in the Venkat and Stewart scheme for faster convergence were proposed in Gross and Stursberg (2013). Still, this DMPC scheme is based on the parallel Jacobi iteration given in (Bertsekas and Tsitsiklis, 1989, chapter 3) and cannot converge to centralized optimality when there are coupled constraints. This is a known barrier in distributed convex optimization for years.

Note that we do not cover the extensive literature on DMPC methods using the *dual decomposition* approach, as it is usually difficult for such methods to guarantee that any intermediate result is feasible for the DMPC problem, which is a key feature in this *primal decomposition* approach.

In this paper, we revisit this DMPC problem from the distributed convex optimization point of view and we propose a new way of organizing the distributed Jacobi iteration that guarantees feasibility; the optimality property can be certified

[1] Support by the Alexander von Humboldt Foundation, by EU via ERC-HIGHWIND (259 166), ITN-TEMPO (607 957), and ITN-AWESCO (642 682) and by DFG via the project "Numerische Methoden zur optimierungsbasierten Regelung zyklischer Prozesse is gratefully acknowledged.

10.1016/j.ifacol.2017.08.744

a posteriori, and in specific cases theoretical proof for convergence to optimality is available. Our main contributions in this paper are:

- We provide a general framework to allow arbitrary decomposition of the centralized variables into local variables, including the case of sharing variables between local problems.
- We provide an *a posteriori* checking method to know whether the fixed point is a centralized optimum.
- For a particular setting of chained link systems, we proposed *a priori* conditions for guaranteeing convergence to centralized optimum.
- The numerical examples show that increasing the number of shared variables (among agents) leads to better performance, and can ultimately achieve centralized optimality.

The paper is organized as follows: the MPC problem is described in Section 2. It is then considered as a convex optimization problem with linear constraints, which will be treated by a distributed iterative Jacobi algorithm proposed in Section 3. Then in Section 4 we provide a theoretical proof for optimality for a special setting. The application of the proposed Jacobi algorithm to distributed MPC is described in Section 5. Section 6 illustrates the algorithm in a numerical example and discusses our observations. Section 7 concludes the paper and outlines our future research.

2. PROBLEM DESCRIPTION

2.1 Coupled dynamical model

Consider a plant consisting of M subsystems. Each subsystem's dynamics is assumed to be influenced directly by only a small number of other subsystems. Let each subsystem be represented by a discrete-time, linear time-invariant model of the form:

$$x_{t+1}^i = \sum_{j=1}^{M} (A_{ij} x_t^j + B_{ij} u_t^j), \tag{1}$$

where $x_t^i \in \mathbb{R}^{n^i}$ and $u_t^i \in \mathbb{R}^{m^i}$ are the states and control inputs of the i-th subsystem at time t, respectively.

Let $x_t = \left[x_t^{1T} \cdots x_t^{MT} \right]^T$ and $u_t = \left[u_t^{1T} \cdots u_t^{MT} \right]^T$ denote the aggregated states and inputs of the full plant at time step t, with dimensions $\sum_{i=1}^{M} n^i$ and $\sum_{i=1}^{M} m^i$ respectively.

The matrices A and B will denote the dynamics matrices of the aggregated system and this linear system is assumed to be stabilizable:

$$A = \begin{bmatrix} A_{11} & \dots & A_{1M} \\ \vdots & & \vdots \\ A_{M1} & \dots & A_{MM} \end{bmatrix}, B = \begin{bmatrix} B_{11} & \dots & B_{1M} \\ \vdots & & \vdots \\ B_{M1} & \dots & B_{MM} \end{bmatrix}.$$

The full (centralized) plant model is thus represented as:

$$x_{t+1} = Ax_t + Bu_t. \tag{2}$$

2.2 Coupled linear constraints

Each subsystem i is assumed to have linear coupled constraints involving only a small number of the other subsystems. We denote the overall coupled constraints at a time t as follows:

$$Cx_t + Du_t \leq d \tag{3}$$

in which C and D are sparse matrices with appropriate sizes.

In addition, the states and control inputs of each subsystem need to satisfy local constraints. We denote the constraints at time step t on the centralized variables as:

$$x_t \in \mathcal{X}, \quad u_t \in \mathcal{U} \tag{4}$$

where $\mathcal{X}$ and $\mathcal{U}$ are closed polytopic sets, containing the origin.

2.3 Centralized MPC problem

The centralized MPC problem is formulated with a finite-horizon and nonnegative convex cost function. Denoting N for the prediction horizon, the centralized MPC problem is then defined as:

$$\begin{aligned} V_t^*(x_t) = \min_{\mathbf{x}_t, \mathbf{u}_t} \ & \sum_{k=0}^{N-1} \ell(x_{t+k}, u_{t+k}) \\ \text{s.t. } & x_{k+1} = Ax_k + Bu_k, \ k = t, \cdots, t+N-1, \\ & Cx_k + Du_k \leq d, \ k = t, \cdots, t+N-1, \\ & x_k \in \mathcal{X}, \ k = t+1, \cdots, t+N \\ & u_k \in \mathcal{U}, \ k = t, \cdots, t+N-1 \\ & x_{t+N} = 0 \\ & x_t = \bar{x}_t, \end{aligned} \tag{5}$$

in which $\bar{x}_t$ is the measured state available at time step t, $\mathbf{x}_t \triangleq [x_t^T, \cdots, x_{t+N}^T]^T$ and $\mathbf{u}_t \triangleq [u_t^T, \cdots, u_{t+N-1}^T]^T$ are the stacked states and control inputs over the horizon, respectively. Note that the function $\ell(\cdot)$ needs to be nonnegative, convex, and have zero value at the origin (e.g. 1-norm or 2-norm cost). The cost function can have couplings between the subsystems.

Remark 2.1

Since the main contribution of this paper is in the distributed optimization algorithm, we use the MPC formulation with a terminal point constraint which is rather conservative but leads to a trivially guaranteed stability when feasibility is maintained. The algorithm presented in this paper will also work with other MPC formulations as far as the stability condition is embedded in the choice of constraints that have a sparse structure. Note that this algorithm does not always guarantee to achieve an optimal solution of the MPC problem. Hence, we need to use MPC settings such that the stability property only relies on feasibility, not on optimality.

One popular way to facilitate the stability property is to extend the prediction horizon N in the MPC formulation. In Keerthi and Gilbert (1988) it was shown that with a prediction horizon N that is long enough to allow a feasible solution to the optimization problem, the closed-loop system is stable. However, increasing N will also lead to more couplings between subsystems in the optimization problem.

2.4 Centralized optimization problem

For the sake of algorithm presentation, we represent the centralized MPC problem (5) in the following optimization form:

$$V^* = \min_{\mathbf{z}} \ f(\mathbf{z}) \tag{6}$$

$$\text{s.t. } \mathcal{A}\mathbf{z} = \mathcal{B} \tag{7}$$

$$\mathcal{C}\mathbf{z} \leq \mathcal{D} \tag{8}$$

with $\mathbf{z} = [\mathbf{x}^T \mathbf{u}^T]^T$; the cost function f and matrices $\mathcal{A}, \mathcal{B}, \mathcal{C}$, and $\mathcal{D}$ are formed appropriately. Note that vectors $\mathcal{B}$ and $\mathcal{D}$ depend on the initial state x_t; however, we drop the argument x_t in order to focus on solving the optimization problem at one

time step. Also note that this is a convex problem, as it is merely a recasting of (5).

The next section deals with the optimization problem (6)–(8) and can be applied to general optimization problems in this form, not only in the field of model predictive control. To simplify notation for applying the distributed optimization method to be presented, we denote n for the size of $\mathbf{z}$, i.e. $\mathbf{z} \in \mathbb{R}^n$.

3. DISTRIBUTED JACOBI ITERATIVE SCHEME

3.1 Decomposition method

In this section, we define the notation for decomposing the centralized variable vector $\mathbf{z}$ into M distributed variable vectors $\mathbf{z}^i$, so that problem (6)–(8) will be solved using a distributed approach with M agents.

We define a decomposition of n scalars in $\mathbf{z}$ into the variables of M subsystems such that each element of a subsystem variable $\mathbf{z}^i \in \mathbb{R}^{n_i}$ is a mapping from an element in $\mathbf{z}$, and every element in $\mathbf{z}$ must belong to at least one subsystem variable. The mathematical expression of the decomposition is given as:

$$\mathbf{z}^i = \begin{bmatrix} z_{i(1)} \cdots z_{i(n_i)} \end{bmatrix}^T : \tag{9}$$

$$\begin{cases} i(j) \in \{1, \cdots, n\}, \forall j \in \{1, \cdots, n_i\} \\ \mathcal{V}^i \triangleq \{i(1), \cdots, i(n_i)\} \\ \mathcal{V}^i \subset \{1, \cdots, n\} \quad , \quad \bigcup_{i=1}^{M} \mathcal{V}^i = \{1, \cdots, n\} \end{cases} \tag{10}$$

Remark 3.1 This notion encapsulates all possibilities for decomposing n scalars into M groups of variables. This also allows overlappings between different groups, i.e. some components can appear in multiple groups. The simplest decomposition is to have no overlapping between any two groups, named *decoupled decomposition*. The other extreme is when each group contains all the variables, i.e. $\mathbf{z}^i = \mathbf{z}$.

For facilitating distributed communications in our approach, we suggest to choose the simplest decomposition that satisfies the following condition: firstly each $\mathbf{z}^i$ includes the variables of the i^{th} physical subsystem, then $\mathbf{z}^i$ also includes all the other variables that have *direct coupling* with any variable of the i^{th} subsystem. A *direct coupling* exists between a pair of variables means that there is at least one row in the constraints (7)–(8) such that the coefficients of those two variables are both nonzero, or there is a nonlinear term coupling those two variables in the cost function (6).

Let $\mathcal{G}$ denote the undirected graph of interactions, which is a set of tuples (i, j) such that either $\mathcal{V}^i \cap \mathcal{V}^j \neq \emptyset$, or there are couplings between variables in $\mathbf{z}^i$ and $\mathbf{z}^j$ in the cost function $f(\cdot)$ or constraints (7)–(8). This graph later shows the communication structure of our scheme.

We define the *neighborhood set* $\mathcal{N}^i$ for each agent i as follows:

$$\mathcal{N}^i = \{j | (i, j) \in \mathcal{G}\} \tag{11}$$

3.2 Overlapping Jacobi iteration

Based on decomposition (9)–(10), we formulate for each $i = 1, \cdots, M$ a local problem as follows:

$$(\mathcal{P}^i): \quad V^i = \min_{\mathbf{z}^i} \; f(\mathbf{z}^i, \tilde{\mathbf{z}}^{-i}) \tag{12}$$

$$\text{s.t.} \;\; \mathcal{A}^i \mathbf{z}^i + \mathcal{A}^{-i} \tilde{\mathbf{z}}^{-i} = \mathcal{B}^i \tag{13}$$

$$\mathcal{C}^i \mathbf{z}^i + \mathcal{C}^{-i} \tilde{\mathbf{z}}^{-i} \leq \mathcal{D}^i \tag{14}$$

in which $\tilde{\mathbf{z}}^{-i} \in \mathbb{R}^{n-n_i}$ denotes given values of variables in $\mathbf{z}$ but outside of $\mathbf{z}^i$, the matrices $\mathcal{A}^i$, $\mathcal{A}^{-i}$, and $\mathcal{B}^i$ are submatrices from $\mathcal{A}$ and $\mathcal{B}$ such that (13) represents all rows of (7) that have a nonzero entry for any component of $\mathbf{z}^i$. In other words, (13) is a compact version of (7) that is relevant for agent i. Similarly, $\mathcal{C}^i$, $\mathcal{C}^{-i}$, and $\mathcal{D}^i$ are submatrices of $\mathcal{C}$ and $\mathcal{D}$ so that (14) is a compact version of (8).

Let $\mathbf{z}_0$ be a feasible point of problem (6)–(8). We solve this problem by an iterative distributed scheme.

At each iteration $p \geq 0$, given the prior, feasible iterate $\mathbf{z}_{(p)}$, each agent i solves problem $(\mathcal{P}^i)$ using $\tilde{\mathbf{z}}^{-i}_{(p)}$ extracted from $\mathbf{z}_{(p)}$. We denote the solution of (12)–(14) as $\mathbf{z}^i_*(\tilde{\mathbf{z}}^{-i}_{(p)})$. Next, we construct $\mathbf{z}^i_{(p+1)}$ from $\mathbf{z}^i_*(\tilde{\mathbf{z}}^{-i}_{(p)})$ and $\tilde{\mathbf{z}}^{-i}_{(p)}$. Note that $\mathbf{z}^i_{(p+1)} \in \mathbb{R}^n$, since it contains all elements of $\mathbf{z}$.

Then, we merge local solutions according to the following convex combination, with a predefined set of $\omega^i, i = 1, \cdots, M$:

$$\mathbf{z}_{(p+1)} = \sum_{i=1}^{M} \omega^i \, \mathbf{z}^i_{(p+1)}, \quad \omega^i \geq 0, \quad \sum_{i=1}^{M} \omega^i = 1 \tag{15}$$

The iteration continues until it reaches either a predefined maximum iteration limit $p_{\max}$, or the result $\mathbf{z}_{(p)}$ converges to a fixed point (to be proved in the next section).

Remark 3.2 While we use $\tilde{\mathbf{z}}^{-i}$ in problem (12)–(14), agent i only needs to communicate with its neighboring agents in $\mathcal{N}^i$ to get meaningful parameters for its problem $(\mathcal{P}^i)$. Due to sparse coupling structure of the constraints, there are many zeros in the matrices $\mathcal{A}^{-i}$ and $\mathcal{C}^{-i}$, which correspond to information in $\tilde{\mathbf{z}}^{-i}$ that is not relevant to problem $(\mathcal{P}^i)$ and do not need to be exchanged with the agent i in communications.

Remark 3.3 This algorithm is a generalization of the cooperative algorithm in (Stewart et al., 2010, Section 3.1.7). It employs only local communications, with better flexibility to choose arbitrary decompositions (as explained in Remark 3.1). We will show in the illustration example that using an *overlapping decomposition* will lead to more optimality than using a *decoupled decomposition*.

Remark 3.4 In (Stewart et al., 2010, Section 5), a hierarchical scheme is also described to deal with coupled constraints. However, if we analyze that scheme thoroughly for the case of many subsystems (i.e. $M \gg 2$), when there are chain-linked coupled constraints such as inequality constraints coupling each two adjacent agents $(i, i+1)$, then such algorithm leads to the formulation that every agent needs to solve a full-size problem since the common variable is indeed the centralized variable. Hence, the algorithm in (Stewart et al., 2010, section 5) is not suitable for distributed optimization.

Remark 3.5 When the optimization solver used for solving local problems (12)–(14) can also provide Lagrange multipliers (which are indeed obtainable for free with e.g. primal-dual interior point methods, for example), then we can use the Lagrange multipliers from local problems to check whether the final result is globally optimal. Thus, while the global optimality is not always guaranteed, we can do an *a posteriori*

check to verify the optimality of the fixed point (to be presented in the next section).

3.3 Properties of the Jacobi algorithm

The following two lemmas establish the feasibility and convergence of the distributed MPC scheme, which is similar to the approaches in Venkat et al. (2008); Stewart et al. (2010):

Lemma 3.6. (Feasibility). Starting from a feasible $\mathbf{z}_0$, every subsequent $\mathbf{z}_{(p)}$ for $p \geq 1$ is also feasible for the centralized optimization problem (12)–(14).

Proof:

Each $\mathbf{z}^i_{(p)}$ obtained from $\mathbf{z}^i_*(\tilde{\mathbf{z}}^{-i}_p)$ and $\tilde{\mathbf{z}}^{-i}_p$ satisfies (13)–(14), and thus it also satisfies (7)–(8). Since these constraints represent a convex constraint set, the convex combination (15) yields a new feasible point for this convex set. □

Lemma 3.7. (Convergence). Denote $V_{(p)} = f(\mathbf{z}_{(p)})$. Then the sequence $\{V_{(p)}\}_{p=1}^{\infty}$ in nonincreasing and converges as $p \to \infty$.

Proof: Using the convexity of $f(\cdot)$ and the the fact that $\mathbf{z}_{(p)}$ is a feasible solution for every $(\mathcal{P}^i)$ which has $\mathbf{z}^i_{(p+1)}$ as an optimizer, we show that $V_{(p)}$ is a nonincreasing sequence:

$$\begin{aligned} V_{(p+1)} &= f\left(\sum_{i=1}^{M} \omega^i \mathbf{z}^i_{(p+1)}\right) \leq \sum_{i=1}^{M} \omega^i f\left(\mathbf{z}^i_{(p+1)}\right) \\ &\leq \sum_{i=1}^{M} \omega^i f\left(\mathbf{z}_{(p)}\right) = f\left(\mathbf{z}_{(p)}\right) = V_{(p)} \end{aligned} \tag{16}$$

Since $V_{(p)}$ has finite value and bounded below, using the monotonicity property, the sequence $V_{(p)}$ converges as $p \to \infty$. □

Note that while this algorithm converges, the fixed point can be suboptimal. However, depending on the decomposition, the performance varies, and the fixed point of the distributed scheme may also be the global optimum. Next, we provide a method to check whether this situation happens.

Lemma 3.8. (A posteriori check of global optimality). Let $\mathbf{z}^\infty$ denote the fixed point of the distributed Jacobi-type algorithm. Then $\mathbf{z}^\infty$ is the global optimizer to problem (6)–(8) if every shared constraint among local problems $(\mathcal{P}^i), i \in \{1, \cdots, M\}$ yields the same Lagrange multiplier across all local problems having that constraint.

Proof:

For $\mathbf{z}^*$ to be the optimizer of the centralized optimization problem, there must exist λ and μ such that the tuple $(\mathbf{z}^*, \lambda, \mu)$ satisfies the following Karush-Kuhn-Tucker (KKT) conditions, where λ and μ respectively denote the Lagrange multiplier vectors associated with inequality constraints (8) and equality constraints (7):

$$\nabla f(\mathbf{z}^*) + \mathcal{C}^T\lambda + \mathcal{A}^T\mu = 0 \tag{17}$$
$$\mathcal{A}\mathbf{z}^* = \mathcal{B} \tag{18}$$
$$\mathcal{C}\mathbf{z}^* \leq \mathcal{D} \tag{19}$$
$$\lambda \geq 0 \cdot \mathbf{1} \tag{20}$$
$$\lambda_1(\mathcal{C}\mathbf{z}^* - \mathcal{D})_1 = 0 \tag{21}$$
$$\vdots \tag{22}$$
$$\lambda_m(\mathcal{C}\mathbf{z}^* - \mathcal{D})_m = 0 \tag{23}$$

in which m denotes the number of rows in matrix $\mathcal{C}$, and $(\mathcal{C}\mathbf{z}^* - \mathcal{D})_k$ is the k-th element of the vector inside the matrix.

The fixed point $\mathbf{z}^\infty$ is the optimizer to all local problems where the value $\tilde{\mathbf{z}}^{-i}$ is taken from $\mathbf{z}^\infty$.

In case every shared constraint has the same multiplier across local problems, we collect all shared multipliers and the distinct multipliers associated with only one local problem, to form the aggregated vectors λ^∞ and μ^∞. Using the KKT condition on the problem (12)–(14), the tuple $(\mathbf{z}^\infty, \lambda^\infty, \mu^\infty)$ must satisfy the following KKT system:

$$\nabla_i f(\mathbf{z}^\infty) + (\mathcal{C}^T\lambda^\infty)^i + (\mathcal{A}^T\mu^\infty)^i = 0 \tag{24}$$
$$(\mathcal{A}\mathbf{z}^\infty)^i = \mathcal{B}^i \tag{25}$$
$$(\mathcal{C}\mathbf{z}^\infty)^i \leq \mathcal{D}^i \tag{26}$$
$$(\lambda^\infty)^i \geq 0 \cdot \mathbf{1}_{m_i} \tag{27}$$
$$(\lambda^\infty)^i_1(\mathcal{C}\mathbf{z}^\infty - \mathcal{D})^i_1 = 0 \tag{28}$$
$$\vdots \tag{29}$$
$$(\lambda^\infty)^i_{n_i}(\mathcal{C}\mathbf{z}^\infty - \mathcal{D})^i_{n_i} = 0 \tag{30}$$

in which the superscript i refers to the selection of the given matrix or column vector to the rows that only related to the variable $\mathbf{z}^i$.

When we concatenate the systems (24)–(30) for $i = 1, \cdots, M$ per each equation and removing the duplicated rows, we get a linear system that is the same as the KKT system (17)–(23) (we only get duplicated rows instead of conflicting rows between different subsystems, thank to the same shared multipliers and the same $\mathbf{z}^\infty$ across all subsystems). Hence, the merging of solutions of local problem to $(\mathbf{z}^\infty, \lambda^\infty, \mu^\infty)$ satisfies the KKT condition of the centralized optimization problem. □

To summarize this section, we have proposed a Jacobi-type algorithm that is applicable to a flexible decomposition method, maintaining feasibility at every iterate. This algorithm converges to a fixed point, and we can do an *a posteriori* check for centralized optimality based on comparing the multipliers between local problems.

4. CONVERGENCE FOR THE SPECIFIC CASE OF CHAIN LINKED SYSTEMS

In this section, we provide the proof for *convergence to global optimality* of the proposed Jacobi algorithm in one specific case. Let us consider the MPC problem which can be recast in the following form:

$$\min_{z_1, \cdots, z_n} \quad f(z_1, \cdots, z_n) \tag{31}$$
$$\text{s.t.} \quad \underline{z_i} \leq z_i \leq \overline{z_i},\ i = 1, \cdots, n \tag{32}$$
$$a_i z_i + b_{i+1} z_{i+1} \leq c_i,\ i = 1, \cdots, n-1 \tag{33}$$

in which the bounds $\underline{z_{i-1}}, \overline{z_{i-1}},\ i = 1, \cdots, n$ and $\{a_i, b_i\}_{j=1\cdots n}$ are constants. For the sake of brevity, we consider in this section each z_i as a scalar.

The Jacobi algorithm is applied, with the following decomposition:

$$\begin{aligned} \mathcal{V}^1 &= \{1, 2\} \\ \mathcal{V}^i &= \{i-1, i, i+1\}, \quad i = 2, \cdots, n-1 \\ \mathcal{V}^n &= \{n-1, n\} \end{aligned}$$

and consequently local problems $(\mathcal{P}^i)$ are constructed as described in Section 3.2.

We provide the following condition that can be verify *a priori*, so that if it holds, the algorithm is guaranteed to converge to optimality:

Theorem 4.1. If for any value of $\tilde{z}^{-1}$, the solution of $(\mathcal{P}^1)$ does not activate the constraint involving z_1 and z_2, and the solution of $(\mathcal{P}^n)$ does not activate the constraint involving z_{n-1} and z_n for any value of $\tilde{z}^{-n}$, then the fixed point of the Jacobi algorithm is globally optimal.

Proof:

Let $\mathbf{z}^\infty \triangleq [z_1^\infty, \cdots, z_n^\infty]^T$ be a fixed point of the algorithm. It is obvious that for $\mathbf{z}^\infty$ to be the fixed point of the algorithm, it must also be the optimal solution of every problem $(\mathcal{P}_\infty^i)$ where $\tilde{\mathbf{z}}^{-i}$ contains values taken from $\mathbf{z}^\infty$.

Suppose that $\mathbf{z}^\infty$ is only suboptimal, i.e. there exists an optimal solution of the centralized problem $\mathbf{z}^* \neq \mathbf{z}^\infty$ and $f(\mathbf{z}^*) < f(\mathbf{z}^\infty)$. Then we will show that there exists at least one subproblem $(\mathcal{P}_\infty^i)$ that does not attain optimality at $\mathbf{z}^\infty$. This will violate the definition that $\mathbf{z}^\infty$ is a fixed point of the algorithm. Hence, the reverse clause must be true, i.e. $\mathbf{z}^\infty$ is a centralized optimal solution. Our strategy for the proof is to start from z_1^∞ and subsequently use the coupled constraints to show that $z_2^\infty, z_3^\infty, \cdots$ must belong to an optimal solution.

Firstly, due to the condition stated in Theorem 4.1, we see that the component z_1^∞ of the fixed point must belong to one centralized optimal solution. We can write $z_1^\infty = z_1^*$.

Next, let we consider problem $(\mathcal{P}_\infty^1)$ which involves z_1, z_2 as variables:

$$
\begin{aligned}
(\mathcal{P}_\infty^1): \quad & \min_{z_1, z_2} \; f(z_1, z_2, z_3^\infty, \cdots, z_n^\infty) && (34)\\
& \text{s.t. } \underline{z_i} \leq z_i \leq \overline{z_i},\ i = 1, 2 && (35)\\
& \quad a_1 z_1 + b_1 z_2 \leq c_1 && (36)\\
& \quad a_2 z_2 + b_2 z_3^\infty \leq c_3 && (37)
\end{aligned}
$$

If the solution of this problem is also optimal for the centralized problem, ie. $(z_1^\infty, z_2^\infty) = (z_1^*, z_2^*)$, then we move on to prove optimality for z_3^∞.

Suppose the contrary, i.e. $z_2^\infty \neq z_2^*$ and $f(z_1^*, z_2^*, \cdot) < f(z_1^*, z_2^\infty, \cdot)$. Then we must have $z_3^\infty \neq z_3^*$, and constraint (37) must be active, i.e. $a_2 z_2^\infty + b_2 z_3^\infty = c_2$. Otherwise, there exists a value $\bar{z}_2 = \alpha z_2^* + (1-\alpha) z_2^\infty$ with $\alpha \in (0, 1]$ such that $(z_1^*, \bar{z}_2)$ is feasible for $(\mathcal{P}_\infty^1)$, and due to convexity of the function f:

$$
\begin{aligned}
f(z_1^*, \bar{z}_2, \cdot) &\leq \alpha f(z_1^*, z_2^*, \cdot) + (1-\alpha) f(z_1^*, z_2^\infty, \cdot)\\
&< f(z_1^*, z_2^\infty, \cdot).
\end{aligned}
$$

Thus, (z_1^*, z_2^∞) is not an optimal solution of $(\mathcal{P}_\infty^1)$.

Now we consider the case (37) is active. From problem $(\mathcal{P}_\infty^2)$, the algorithm will branch out two cases: $a_3 z_3^\infty + b_3 z_4^\infty = c_3$ or $a_3 z_3^\infty + b_3 z_4^\infty < c_3$. In the case where this constraint is inactive, we also use convexity to find a better point $(z_1^*, \bar{z}_2, \bar{z}_3)$ lying between (z_1^*, z_2^*, z_3^*) and $(z_1^*, z_2^\infty, z_3^\infty)$ such that:

$$
\begin{aligned}
a_3 \bar{z}_3 + b_3 z_4^\infty &\leq c_3\\
a_2 \bar{z}_2 + b_2 \bar{z}_3 &\leq c_2\\
f(z_1^*, \bar{z}_2, \bar{z}_3, \cdot) &< f(z_1^*, z_2^\infty, z_3^\infty, \cdot)
\end{aligned}
$$

which shows that $(z_1^*, z_2^\infty, z_3^\infty)$ is not an optimal solution of $(\mathcal{P}_\infty^2)$.

Similarly, we can show that there must be a sequence of active constraints in order to allow $z_2^\infty \neq z_2^*$:

$$
\begin{aligned}
a_4 z_4^\infty + b_4 z_5^\infty &= c_4\\
&\vdots\\
a_{i-1} z_{i-1}^\infty + b_{i-1} z_i^\infty &= c_{i-1}
\end{aligned}
$$

For any index j such that $a_j z_j^\infty + b_j z_{j+1}^\infty < c_j$, we can then find a feasible point $(z_1^*, \bar{z}_2, \cdots, \bar{z}_j)$ that lowers the cost function, hence, $(z_{j-2}^\infty, z_{j-1}^\infty, z_j^\infty)$ is not an optimal solution of $(\mathcal{P}_\infty^{j-1})$.

Therefore, in order to the optimality of problems $(\mathcal{P}_\infty^i)$ up to $i = n-1$ at the point z^∞, the sequence of active contraints must be extended until $i = n-1$, i.e we have $a_{n-1} z_{n-1}^\infty + b_{n-1} z_n^\infty = c_n$. However, this equality violate the condition in Theorem 4.1 that such constraint in $(\mathcal{P}_\infty^n)$ must be is inactive. This shows that there is no $\mathbf{z}^* \neq \mathbf{z}^\infty$ such that $f(\mathbf{z}^*) < f(\mathbf{z}^\infty)$. This concludes the proof. □

5. APPLICATION TO DISTRIBUTED MPC

In this section, we focus on applying the iterative algorithm in the previous section to implement a closed-loop state-feedback distributed MPC system, following the formulation (5) of the centralized MPC problem.

5.1 Defining the decomposition

Firstly, based on the nature of couplings in the MPC problem, we define a decomposition so that there are M agents for carrying out the distributed scheme during online operation.

Recall that in Section 2.1, the states x^i and inputs u^i are defined. If we choose the decomposition such that $\mathbf{z}^i$ only contains the predicted variables of x^i and u^i, then we have a non-overlapping decomposition (cf. Remark 3.1).

Following the description in Section 3.1, a graph of interactions $\mathcal{G}$ is constructed. In this graph, we define the *neighborhood of* i, denoted by $\mathcal{N}^i$, as the set of nodes that have a direct link with node i in the graph. So $\mathcal{N}^i$ encompasses subsystems that have direct couplings with subsystem i (either in the cost function, dynamical constraints, or other constraints).

We can define other overlapping decompositions intuitively, so that the group of variables for a subsystem forms an *r-step extended neighborhood*. The notion of the *r-step extended neighborhood* for a subsystem i, denoted by $\mathcal{N}_r^i$, is the set that contains all nodes that can be reached from node i in not more than r links. Specifically, $\mathcal{N}_r^i$ is defined recursively as the union of subsystems in the neighborhoods of all subsystems in $\mathcal{N}_{r-1}^i$:

$$
\mathcal{N}_r^i = \bigcup_{j \in \mathcal{N}_{r-1}^i} \mathcal{N}^j, \tag{38}
$$

where $\mathcal{N}_1^i := \mathcal{N}^i$.

5.2 Using online algorithms to generate feasible points

We assume that at the first sampling time $(t = 0)$, with a given initial state x_0, a feasible solution $(\tilde{\mathbf{x}}_0, \tilde{\mathbf{u}}_0)$ to problem (5) is available. Hereby we describe the scheme for online distributed MPC, starting from $t = 0$.

At sampling time t, the state x_t is available, and then an optimization problem (5) needs to be solved. With a prior feasible candidate $(\tilde{\mathbf{x}}_t, \tilde{\mathbf{u}}_t)$, we apply the Jacobi-type algorithm for solving the centralized problem in a distributed fashion.

The algorithm can stop after $p \leq p_{\max}$ steps, result in the solution denoted as $\mathbf{z}_{(p|t)}$. The subsystems then take the result $\mathbf{z}^i_{(p|t)}, i = 1, \cdots, M$ and parse them to their own sequences $x^i_{t+1|t}, \cdots, x^i_{t+N|t}$ and $u^i_t, \cdots, u^i_{t+N-1|t}$. Then each control action $u^i_t, i = 1, \cdots, M$ is applied to the plant at sampling instant t.

By the end of sampling step t, the state of the plant is $x_{t+1} = \left[{x^1_{t+1}}^T, \cdots, {x^M_{t+1}}^T\right]^T$ (in this theoretical work we assume state feedback without uncertainties and disturbances). The next feasible sequence for the MPC problem at time step $t+1$ is constructed as follows: the zero input $u^i_{t+N} = 0$ and state $x^i_{t+N+1} = 0$ are respectively appended to each local prediction control sequence and prediction state sequence; then these two sequences are shifted one step to the right, they are used as the starting feasible candidate for time step $t+1$. It is obvious that the centralized sequences $(\tilde{\mathbf{x}}_{t+1}, \tilde{\mathbf{u}}_{t+1})$, combined of all local sequences, are feasible for problem (5) at time step $t+1$.

Note that the centralized vectors mentioned above are only used for the sake of expressing the recursive feasibility. Indeed, during the whole process of implementation, we do not need to construct the centralized vectors. Each agent only needs to communicate with its neighbors and deals with its local vectors.

6. NUMERICAL EXAMPLE

In this section, we illustrate the application of the proposed algorithm to a problem involving a chain of coupled oscillators. The problem setup consists of M oscillators that can move only along the vertical axis, and that are coupled by springs that connect each oscillator with its two closest neighbors. An exogenous vertical force will be used as the control input for each oscillator. The setup is shown in Figure 1.

Each oscillator is considered as one subsystem. Let the superscript i denote the index of the oscillators. The dynamics equation of oscillator i is then defined as

$$ma^i = k_1 p^i - f_s v^i + k_2(p^{i-1} - p^i) + k_2(p^{i+1} - p^i) + F^i \quad (39)$$

where p^i, v^i, and a^i denote the position, velocity, and acceleration of oscillator i, respectively, the control force exerted at oscillator i is F^i, and the following parameters are identical for all oscillators:

k_1: stiffness of vertical spring at each oscillator

k_2: stiffness of springs that connect the oscillators

m: mass of each oscillator

f_s: friction coefficient of movements

From some nonzero initial state, the system needs to be stabilized subject to the constraints:

$$\left|p^i - p^{i-1}\right| \leq 2, \quad i = 2, ..., M-1 \quad (40)$$

$$\left|p^i\right| \leq 2, \quad \forall i \quad (41)$$

We use quadratic cost functions for the MPC problem:

$$\ell(x_k, u_k) = \sum_{i=1}^{M} \|x^i_k\|_{Q_i} + \|u^i_k\|_{R_i} \quad (42)$$

with Q_i, R_i positive definite matrices. This formulation leads to a convex, decoupled cost function.

Based on dynamical couplings and constraint couplings, the neighborhood of each subsystem inside the chain is defined to contain itself and its two closest neighbors $\mathcal{N}^i = \{i-1, i, i+1\}, i = 2, ..., M-1$, while for the two ends $\mathcal{N}^1 = \{1, 2\}$ and $\mathcal{N}^M = \{M, M-1\}$. We define the state vector as $x^i = [p^i, v^i]^T$, and the input as $u^i = F^i$. The discretized dynamics with sampling time T_s, obtained using the forward Euler discretization method, is represented by the following matrices:

$$A_{ij} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \forall j \notin \mathcal{N}^i$$

$$A_{i,i-1} = \begin{bmatrix} 0 & 0 \\ T_s k_2 & 0 \end{bmatrix}, \text{for } i = 2, ..., M$$

$$A_{ii} = \begin{bmatrix} 1 & T_s \\ T_s(k_1 - 2k_2) & 1 - T_s f_s \end{bmatrix}, \text{for } i = 1, ..., M$$

$$A_{i,i+1} = \begin{bmatrix} 0 & 0 \\ T_s k_2 & 0 \end{bmatrix}, \text{for } i = 1, ..., M-1$$

$$B_{ij} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \forall j \neq i, \quad B_{ii} = \begin{bmatrix} 0 \\ T_s \end{bmatrix}, \text{for } i = 1, ..., M$$

The following parameters were used in the simulation example:

$$k_1 = 0.4, \quad k_2 = 0.3$$
$$f_s = 0.4, \quad T_s = 0.05, \quad m = 1$$
$$M = 40, \quad N = 20$$
$$Q_i = \begin{bmatrix} 100 & 0 \\ 0 & 0 \end{bmatrix}, \quad R_i = 10$$

Firstly, we use the overlapping decomposition in which the local variable vector $\mathbf{z}^i$ includes all variables of subsystems inside the *one-step neighborhood* $\mathcal{N}^i_1$. Starting from the same feasible initial state, we apply the distributed MPC algorithm with $p_{\max} = 2$, 20, and 100. The results are compared to the exact solution obtained from the centralized MPC approach (obtained using any solver that gives numerical solution that can be considered *exact*). The results indicate that all states of the 40 subsystems are stabilized. Figure 2 shows the evolution of the overall cost achieved by distributed MPC compared to the cost of the centralized approach. We can see that the difference is reduced by choosing a larger $p_{\max}$ value. Thank to the use of *overlapping decomposition*, the distributed solution converges to the *centralized optimum*. When we compare the Lagrange multipliers of local problems for the results with $p_{\max} = 100$, taking into account the slight errors due to computer precision, we can also verify that Lemma 3.8 holds for every MPC time step. Indeed, we can use the theoretical condition tp verify optimality, similar to Section 4 and taking into account the difference of coupled constraint forms between (40) and (33).

Another test is to compare the effect of different ways of decomposing the problem. We let each local problem in the distributed MPC algorithm optimize over the inputs of subsystems in a larger neighborhood, meaning there are more overlappings in the decomposition. Figure 3 illustrates the effect of optimizing in each subproblem over an r-step extended neighborhood, with $r \in \{1, 5, 10\}$. Note that due to the coupled constraints (40) between any pair of subsystems, using *non-overlapping decomposition* will not guarantee feasibility, hence we must use *overlapping decomposition* with the minimum radius of neighborhood $r = 1$. Fixing the number of maximum iterations in the distributed Jacobi iteration to $p_{\max} = 2$, we observe a steady improvement in performance until the increased neigh-

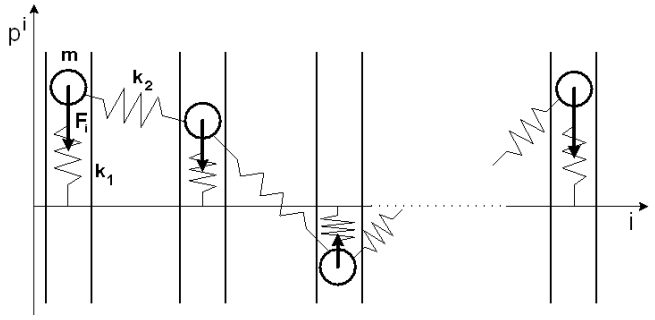

Figure 1. Setup of coupled oscillators

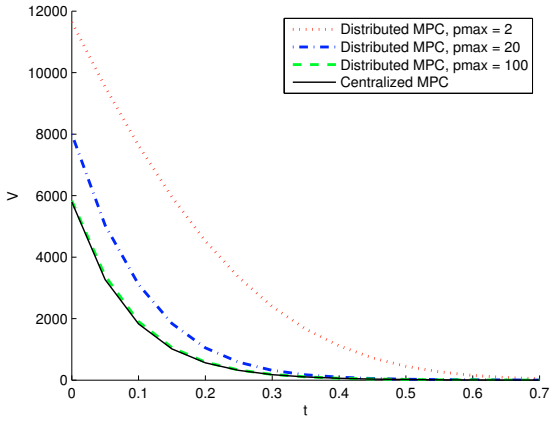

Figure 2. Time evolution of the cost value of the centralized MPC algorithm in comparison with the distributed MPC algorithm using $p_{\max} = 2$, $p_{\max} = 20$ and $p_{\max} = 100$.

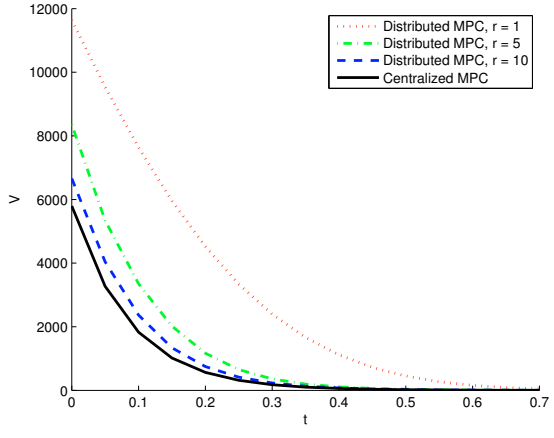

Figure 3. Time evolution of the global cost value of distributed MPC algorithms with different radii of neighborhood to be optimized by one local controller.

borhood of each subsystem covers essentially all other subsystems and we end up with considering the centralized problem. It is worth noting that a more overlapping decomposition leads to increased requirements on information exchanges as well as bigger problem sizes for local problems; hence this involves a trade-off between performance versus communications and computations.

7. CONCLUSIONS

We have presented a Jacobi algorithm for solving distributed convex optimization problems that is applicable to distributed model predictive control. The optimization problem to be solved should have sparse couplings, either in the cost or the constraint functions. Consequently, the DMPC scheme is able to deal with linear time-invariant systems with general linear coupled dynamics and convex coupled constraints. Recursive feasibility and stability are guaranteed, and we are flexible to choose a decomposition method that is favorable to the communication structure and computational capabilities of the local controllers. It is shown that enlarging the overlappings in decomposition could improve the performance, and that the DMPC solution can converge to the centralized one in practice, although in theory only suboptimality could be proved. We provide an *a posteriori* checking method to verify whether the converging result is centralized optimal. Furthermore, we also propose a condition for a particular chain linked system structure so that the convergence of distributed Jacobi algorithm to the centralized optimality could be checked *a priori*.

REFERENCES

Bertsekas, D.P. and Tsitsiklis, J.N. (1989). *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Upper Saddle River, NJ.

Camponogara, E., Jia, D., Krogh, B., and Talukdar, S. (2002). Distributed model predictive control. *IEEE Control Systems Magazine*, 22(1), 44–52.

Gross, D. and Stursberg, O. (2013). On the convergence rate of a Jacobi algorithm for cooperative distributed MPC. In *52nd IEEE Conference on Decision and Control*, 1508–1513. Florence, Italy.

Jia, D. and Krogh, B. (2001). Distributed model predictive control. In *Proceedings of the American Control Conference*, 2767–2772.

Keerthi, S.S. and Gilbert, E.G. (1988). Optimal infinite-horizon feedback control laws for a general class of constrained discrete-time systems: stability and moving-horizon approximations. *Journal of Optimization Theory and Applications*, 57(2), 265–293.

Rawlings, J.B. and Mayne, D.Q. (2009). *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, WI.

Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control - A review. *Journal of Process Control*, 19(5), 723–731.

Stewart, B.T., Venkat, A.N., Rawlings, J.B., Wright, S.J., and Pannocchia, G. (2010). Cooperative distributed model predictive control. *Systems & Control Letters*, 59, 460–469.

Stewart, B.T., Wright, S.J., and Rawlings, J.B. (2011). Cooperative distributed model predictive control for nonlinear systems. *Journal of Process Control*, 21(5), 698–704. Special Issue on Hierarchical and Distributed Model Predictive Control.

Venkat, A., Hiskens, I., Rawlings, J., and Wright, S. (2008). Distributed MPC strategies with application to power system automatic generation control. *IEEE Transactions on Control Systems Technology*, 16(6), 1192–1206.

Venkat, A., Rawlings, J., and Wright, S. (2005). Stability and optimality of distributed model predictive control. In *44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05.*, 6680–6685.