Parsimonious Cooperative Distributed MPC for Tracking Piece-Wise Constant Setpoints

Matteo Razzanelli [*] **Gabriele Pannocchia** [**]

[*] *Department of Information Engineering, Univ. of Pisa, Italy*
(e-mail: matteo.razzanelli@ing.unipi.it)
[**] *Department of Civil and Industrial Engineering, Univ. of Pisa, Italy*
(e-mail: gabriele.pannocchia@unipi.it). Corresponding author.

Abstract: Distributed Model Predictive Control refers to a class of predictive control architectures in which a number of local controllers manipulate a subset of inputs to regulate a subset of outputs composing the overall system. These controllers may cooperate to find an optimal control sequence that minimizes a global cost function, as in the case of Cooperative Distributed Model Predictive Control (CD-MPC). In this paper two linear CD-MPC algorithms for tracking are proposed. The aim of these controllers is to drive the outputs of the overall system to any admissible piece-wise constant set-point, satisfying input and state constraints. However, in the available literature this result is achieved by using a set of centralized variables that keep track of the global state of the system. In contrast, we develop novel CD-MPC approaches for tracking that rely on "as local as possible" information instead of the plant-wide information flow. These new control strategies reduce the required communication overhead, local computational demands, and are more scalable than CD-MPC algorithms available in the literature. We illustrate the main characteristics and benefits of the proposed approaches by means of a multiple evaporator process example.

Keywords: MPC, reference tracking, cooperative distributed control, large-scale systems

1. INTRODUCTION

Model predictive control (MPC) is by far the most successful advanced control technique applied in the process industries, capable of providing controllers that ensure closed-loop stability, constraint satisfaction, and robustness for multivariable (linear and nonlinear) systems (Maciejowski, 2000; Rawlings and Mayne, 2009; Pannocchia et al., 2011). Typical theoretical results on MPC consider the regulation problem. Nonzero piece-wise constant setpoints can be accounted for by solving a target calculation problem and shifting the origin of the system accordingly. Depending on the extent of the required setpoint change, feasibility issues may arise. To overcome this problem, in (Limon et al., 2008) a centralized MPC for tracking applicable to constrained linear systems is proposed, which ensures nominal recursive feasibility and stability of the closed-loop system under any change of the set-point.

Large-scale systems (e.g. industrial processing plants, power generation networks, etc.) usually comprise several interconnected units which may exchange material, energy and information streams. Thus, industrial control systems are often *decentralized*. Between centralized and decentralized strategies (Bemporad and Barcelli, 2010; Riverso et al., 2013), *distributed* control algorithms preserve topology and flexibility of decentralized controllers and may offer nominal closed-loop stability guarantees. In *non-cooperative* distributed control, each subsystem controller anticipates the effect of interactions only locally, i.e. to optimize a local objective function (Farina and Scattolini, 2012; Betti et al., 2014). However, if these interactions are strong, non-cooperative control can destabilize the plant and performance can be poorer than decentralized control. Alternatively, *cooperative* distributed model predictive control (Rawlings and Mayne, 2009; Stewart et al., 2010) guarantees nominal closed-loop stability and convergence to the centralized optimal performance by requiring each subsystem to consider the effect of local control actions on all other subsystems. So, each local controller optimizes the same plant-wide objective function over its local inputs. In (Ferramosca et al., 2013) a Cooperative Distributed linear Model Predictive Control (CD-MPC) strategy to track changing set-points, applicable to any finite number of subsystems, is presented. This paper extends the formulation in (Ferramosca et al., 2013) to develop novel cooperative distributed MPC approaches for tracking that reliy on information that is "as local as possible".

Notation. The field of reals is denoted by $\mathbb{R}$. The identity matrix is denoted by I, and the zero matrix is denoted by 0. Dimensions of those matrices are omitted when they can be easily inferred from the context or indicated as subscripts. For any $x \in \mathbb{R}^n$, the symbol $\|x\|$ denotes the 2-norm. Given a positive semi-definite matrix $Q \in \mathbb{R}^{n\times n}$, we denote $\|x\|_Q^2 = x^T Qx$. The superscript ^{0} denotes an optimal cost or vector. Operators $\text{diag}\{T_1,\ldots,T_M\}$ and $\text{hor}\{T_1,\ldots,T_M\}$ represent block diagonal and the horizontal concatenation, respectively.

2. PRELIMINARIES

2.1 Overall system and subsystems

This paper focuses on discrete-time, linear, time-invariant systems (DLTI) in the form:

$$\begin{aligned} x^+ &= Ax + Bu \\ y &= Cx \end{aligned} \tag{1}$$

Copyright © 2016 IFAC

in which $x \in \mathbb{R}^n$ and $x^+ \in \mathbb{R}^n$ are the current and successor state, $u \in \mathbb{R}^m$ is the manipulated input, $y \in \mathbb{R}^p$ is the controlled output. Without loss of generality, we assume that the overall system (1) can be represented as the union of M DLTI subsystems. The evolution of the i-th subsystem ($i \in \{1,\dots,M\}$) is given by:

$$\begin{aligned} x_i^+ &= A_i x_i + B_i u_i + \sum_{j \in \mathcal{N}_i} B_{ij} u_j \\ y_i &= C_i x_i \end{aligned} \tag{2}$$

where $\mathcal{N}_i$ is the set of neighbors of subsystem i, and $x_i \in \mathbb{R}^{n_i}$, $x_i^+ \in \mathbb{R}^{n_i}$, $u_i \in \mathbb{R}^{m_i}$, $y_i^+ \in \mathbb{R}^{p_i}$. Note that it is not necessary to consider in (2) state interactions among subsystems, because the "local" state x_i may be augmented (if necessary) to include other required state interaction terms.

Assumption 1. For each subsystem i: the state is measurable at each decision time and the pair (A_i, B_i) is controllable.

We remark that Assumption 1 implies that (A,B) is stabilizable.

2.2 Useful reminders of graph theory

In order to represent and analyze the different subsystems, it is useful to recall a few concepts from graph theory. A graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ is composed by a finite set of vertices (or nodes) $\mathscr{V}$ and a set of edges (or lines) $\mathscr{E} \subset \mathscr{V} \times \mathscr{V}$ that connect pairs of vertices. A graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ is *directed* if $\mathscr{E}$ is composed by oriented edges between the two nodes. Edge (v_i, v_j) is an edge from v_i to v_j, i.e. v_j is the edge head and v_i is the edge tail. Given a directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, the *inlet star* and the *outlet star* of node v_i are, respectively, the following sets:

$$S_i^{IN} = \{v_j \in \mathscr{V} \,|\, (v_j, v_i) \in \mathscr{E}\} \quad S_i^{OUT} = \{v_j \in \mathscr{V} \,|\, (v_i, v_j) \in \mathscr{E}\}$$

We simplify the notation, indicating with i the generic node v_i, and with j the generic node v_j belonging to inlet star of i, $\left(j \in S_i^{IN}\right)$, or to outlet star of i, i.e. $\left(j \in S_i^{OUT}\right)$. Each subsystem i in (2) can be seen as a node of a graph: the set of its neighbors $\mathcal{N}_i$ coincides with its inlet star $\left(j \in S_i^{IN}\right)$, whereas S_i^{OUT} is the set of subsystems of which subsystem i is neighbor. The goal of this work is to manage large-scale systems, with multiple inputs, multiple outputs, and state and input constraints. Moreover, each subsystem, may influence each other.

2.3 Centralized MPC: regulation and tracking

We first recall the basics of MPC in its centralized form.

Centralized regulation. Consider the DLTI system (1) such that $(x, u) = (0, 0)$ is an equilibrium point. Let N be a positive horizon length. Suppose that the system is subject to state and input constraints:

$$x(k) \in \mathbb{X}, \quad u(k) \in \mathbb{U}$$

Given the current state, x, and a finite-horizon input sequence $\mathbf{u} = \{u(0), \dots, u(N-1)\}$ we define the cost function as:

$$\begin{aligned} V(x, \mathbf{u}) &= \sum_{k=0}^{N-1} \ell\left(x(k), u(k)\right) + V_f\left(x(N)\right) \quad \text{s.t.} \\ x(0) &= x \\ x(k+1) &= Ax(k) + Bu(k) \quad k = 0, \dots, N-1 \end{aligned} \tag{3}$$

in which $\ell(\cdot)$ is the stage cost function. If $\ell(\cdot)$ is positive definite, goal of the regulator is to steer the state to the origin. Usually quadratic stage and terminal costs are considered respectively in the form:

$$\ell(x, u) = \frac{1}{2}\left(x^T Q x + u^T R u\right) \quad V_f(x) = \frac{1}{2} x^T P x$$

with Q, R and P positive definite matrices. We can define the finite horizon optimal control problem (FHOCP) as

$$\mathbb{P}(x): \qquad V^0(x) = \min_{\mathbf{u}} \{V(x, \mathbf{u}) \mid \mathbf{u} \in \mathscr{U}_N(x)\} \tag{4}$$

where: $\mathscr{U}_N(x) = \left\{\mathbf{u} \mid u(k) \in \mathbb{U}, x(k) \in \mathbb{X}, x(N) \in \mathbb{X}_f \subseteq \mathbb{X}\right\}$.

$\mathbb{P}(x)$ in (4) provides $\mathbf{u}^0(x)$ as optimal input sequence, associated with a corresponding optimal state sequence $\mathbf{x}^0(x) = \left\{x^0(0) = x, x^0(1), \dots, x^0(N)\right\}$. Finally, we recall that $\mathbb{P}(x)$ can be posed as a Quadratic Program (QP) and solved numerically.

Centralized tracking. It is a common objective to drive the outputs of a system (or a subset of them) to a desired target (y_t) other than the origin. Clearly, this also means that input and state vectors have to reach an equilibrium, in general, different from the origin. Let (x_s, u_s, y_s) denote the steady-state equilibrium triple of state, input and output. From (1), the following relation must hold:

$$\begin{bmatrix} A-I & B & 0 \\ C & 0 & -I \end{bmatrix} \begin{bmatrix} x_s \\ u_s \\ y_s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5}$$

If we define

$$V_{ss}(y_s, y_t) = \|y_s - y_t\|_T^2 \tag{6}$$

with T positive definite diagonal matrix, we obtain an equilibrium triple in which the output is as close as possible to the desired target, while state and input constraints are fulfilled, by solving the following problem:

$$\min_{x_s, u_s, y_s} \quad V_{ss}(y_s, y_t) \quad \text{s.t. (5) and } x_s \in \mathbb{X}, \quad u_s \in \mathbb{U} \tag{7}$$

The above is often referred to as a steady-state target optimizer (SSTO) problem, and we remark that (7) is also a QP. Given $\left(x_s^0, u_s^0, y_s^0\right)$, solution to problem (7), we can define the deviation variables as $\tilde{x} = x - x_s^0$ and $\tilde{u} = u - u_s^0$. Then, we solve the FHOCP in deviation variables:

$$\mathbb{P}(\tilde{x}): \qquad V^0(\tilde{x}) = \min_{\tilde{\mathbf{u}}} \left\{V\left(\tilde{x}(0), \tilde{\mathbf{u}}\right) \middle| \tilde{\mathbf{u}} \in \tilde{\mathscr{U}}_N(\tilde{x})\right\} \tag{8}$$

where: $\tilde{\mathscr{U}}_N(\tilde{x}) = \{\tilde{\mathbf{u}} \mid \tilde{u}(k) + u_s^0 \in \mathbb{U}, \tilde{x}(k) + x_s^0 \in \mathbb{X}, \tilde{x}(N) \in \mathbb{X}_f\}$. The receding horizon control law uses the first element of the optimal sequence $\tilde{\mathbf{u}}^0(x)$solution to (8), so the input will be $u = \tilde{u}^0\left(\tilde{x}(0)\right) + u_s^0$. In some tracking formulations (Limon et al., 2008) the steady-state problem (7) can be embedded into problem $\mathbb{P}(\tilde{x})$, resulting in a single-layer MPC structure.

2.4 Distributed MPC: regulation and tracking

Regulation and tracking MPC algorithms can be designed for each subsystem i in a cooperative fashion, so that a plant-wide cost function can be minimized.

Cooperative regulation. As in centralized MPC, for each subsystem we consider the following cost function:

$$\begin{aligned} V_i\left(x_i, \mathbf{u}_i, \left\{\mathbf{u}_j\right\}_{j \in \mathcal{N}_i}\right) &= \sum_{k=0}^{N-1} \ell_i\left(x_i(k), u_i(k)\right) + V_{fi}\left(x_i(N)\right) \\ \text{s.t.} \quad x_i(0) &= x_i \\ x_i(k+1) &= A_i x_i(k) + B_i u_i(k) + \sum_{j \in \mathcal{N}_i} B_{ij} u_j(k) \end{aligned} \tag{9}$$

in which

$$\ell_i(x_i,u_i) = \tfrac{1}{2}(x_i^T Q_i x_i + u_i^T R_i u_i), \qquad V_{fi}(x_i) = \frac{1}{2} x_i^T P_i x_i$$

with $Q_i \in \mathbb{R}^{n_i}$, $R_i \in \mathbb{R}^{m_i}$ and $P_i \in \mathbb{R}^{n_i}$, positive definite matrices. We remark that $V_i(\cdot)$ depends also on neighbors' input sequences, $\mathbf{u}_j$ for $j \in \mathcal{N}_i$, due to interactions among subsystems, as given in (2). In cooperative distributed MPC (Rawlings and Mayne, 2009; Scattolini, 2009; Pannocchia, 2014), each subsystem optimizes the global objective function defined as

$$V(x, \mathbf{u}) = \sum_{i=1}^{M} V_i\left(x_i, \mathbf{u}_i, \left\{\mathbf{u}_j\right\}_{j\neq i}\right) \tag{10}$$

We notice that the global cost function (10) is equivalent to that of the centralized problem (3) with weight matrices $Q = \text{diag}\{Q_1, \ldots, Q_M\}$, $R = \text{diag}\{R_1, \ldots, R_M\}$, $P = \text{diag}\{P_1, \ldots, P_M\}$. So we define the FHOCP for each subsystem

$$\begin{aligned} \mathbb{P}_i\left(x, \left\{\mathbf{u}_j\right\}_{j\neq i}\right): \quad & \min_{\mathbf{u}_i} V(x, \mathbf{u}) \quad \text{s.t.} \\ & \mathbf{u}_i \in \mathscr{U}_i\left(x, \left\{\mathbf{u}_j\right\}_{j\neq i}\right) \\ & x(N) \in \mathbb{X}_f \subseteq \mathbb{X} \end{aligned} \tag{11}$$

in which

$$\mathscr{U}_i\left(x, \left\{\mathbf{u}_j\right\}_{j\neq i}\right) = \left\{\mathbf{u}_i \mid u_i(k) \in \mathbb{U}_i, x(k) \in \mathbb{X}\right\}$$

We remark that each subsystem solves problem (11) for a known and fixed value of all other subsystem input sequences. If time allows it, this process can be repeated performing cooperative iterations as detailed next. Let $\mathbf{u}_i^{[q-1]}$ be the known value of subsystem i input sequence at the q-th cooperative iteration ($q = 1$, at the beginning of this iterative process), and let $\mathbf{u}_i^0$ be the solution to problem $\mathbb{P}_i\left(x, \left\{\mathbf{u}_j^{[q-1]}\right\}_{j\neq i}\right)$ in (11). Then the input sequence of subsystem i for the next cooperative iteration is defined as a convex combination of the new and old values, i.e.: $\mathbf{u}_i^{[q]} = w_i \mathbf{u}_i^0 + (1-w_i)\mathbf{u}_i^{[q-1]}$, in which $w_i > 0$ such that $\sum_{i=1}^{M} w_i = 1$. Cooperative iterations are typically performed until convergence, i.e. when $\left\|\mathbf{u}_i^{[q]} - \mathbf{u}_i^{[q-1]}\right\|$ is less than a given tolerance, or until a maximum number of iterations is reached. Then, the first component of computed input is sent to each subsystem in the usual receding horizon fashion.

Cooperative tracking. Problem (11) described above can be applied to tracking or can be integrated with dynamic optimization layer as in (Ferramosca et al., 2013). In that work, an *artificial* equilibrium triple (x_s, u_s, y_s) of the overall system (1), i.e. satisfying (5), is added as decision variable. Consequently, the global cost function is modified by adding a term that penalizes the deviation of y_s from y_t, as follows:

$$\begin{aligned} V_t(x, \mathbf{u}, x_s, u_s, y_s) = \sum_{k=0}^{N-1} & \ell(x(k) - x_s, u(k) - u_s) + \\ & V_f(x(N) - x_s) + V_{ss}(y_s, y_t) \\ \text{s.t.} \quad & x(0) = x \\ & x(k+1) = Ax(k) + Bu(k) \end{aligned} \tag{12}$$

in which $V_{ss}(\cdot)$ is defined in (6). The FHOCP to be solved by each subsystem i reads as follows:

$$\begin{aligned} \mathbb{P}_i\left(x, \left\{\mathbf{u}_j\right\}_{j\neq i}\right): \quad & \min_{\mathbf{u}_i, x_s, u_s, y_s} V_t(x, \mathbf{u}, x_s, u_s, y_s) \quad \text{s.t.} \\ & \mathbf{u}_i \in \mathscr{U}_i\left(x, \left\{\mathbf{u}_j\right\}_{j\neq i}\right) \\ & (x(N), y_s) \in \Omega \end{aligned} \tag{13}$$

where Ω is an admissible polyhedral invariant set for tracking for system (1), as explained in (Ferramosca et al., 2013). We also point out that in problem (13) solved by (Ferramosca et al., 2013), the triple (x_s, u_s, y_s) is parameterized by y_s only, which represents together with $\mathbf{u}_i$ the actual decision variable.

3. PROPOSED METHODOLOGY

The proposed method is based on a single optimization layer as described in § 2.4. Differently from the general approach used in cooperative algorithms (Rawlings and Mayne, 2009; Ferramosca et al., 2013; Pannocchia, 2014), in the proposed method each local controller does not have to know the overall system state but only a part of this one that is strictly necessary to achieve the desired global tracking goal. To this aim we need to exploit carefully the interactions among subsystems.

3.1 The augmented system

Each subsystem i is influenced by the inputs of its *inlet star*, as described in (2) and its input influences the subsystems of its *outlet star*, along with the inputs of their inlet stars (cfr. §2.3). The evolution of its own state x_i and of the states of subsystems belonging to its *outlet star* are given by:

$$x_i^+ = A_i x_i + B_i u_i + \sum_{k \in S_i^{IN}} B_{ik} u_k \tag{14}$$

$$x_j^+ = A_j x_j + B_{ji} u_i + \left(B_j u_j + \sum_{k \in S_j^{IN} \setminus \{i\}} B_{jk} u_k\right), \quad j \in S_i^{OUT}$$

The evolution of the states of the remaining subsystems can be written as:

$$x_j^+ = A_j x_j + \left(B_j u_j + \sum_{k \in S_j^{IN}} B_{jk} u_k\right), \quad j \notin S_i^{OUT} \tag{15}$$

It is therefore clear that each subsystem i should only consider the evolution of subsystems reported in (14), because those reported in (15) are independent of u_i. We can define a new set representing the inlet star of the augmented subsystem (14):

$$\mathbb{S}_i^{IN} \leftarrow S_i^{IN} \cup S_i^{OUT} \cup \left(\bigcup_{j \in S_i^{OUT}} S_j^{IN} \setminus \{i\}\right) \tag{16}$$

Note that by definition, $i \notin \mathbb{S}_i^{IN}$. Then, defining the following stacked vectors and matrices:

$$\bar{x}_i = \begin{bmatrix} x_i \\ [x_j]_{j \in S_i^{OUT}} \end{bmatrix}, \quad \bar{u}_i = [u_k]_{k \in \mathbb{S}_i^{IN}}, \quad \bar{y}_i = \begin{bmatrix} y_i \\ [y_j]_{j \in S_i^{OUT}} \end{bmatrix}$$

$$\bar{A}_i = \text{diag}\left\{A_i, \{A_j\}_{j \in S_i^{OUT}}\right\}, \quad \bar{B}_i = \begin{bmatrix} B_i \\ [B_{ji}]_{j \in S_i^{OUT}} \end{bmatrix}$$

$$\bar{B}_i^{IN} = \begin{bmatrix} \text{hor}\{B_{ik}\}_{k \in \mathbb{S}_i^{IN}} \\ \text{hor}\{B_{jk}\}_{j \in S_i^{OUT}, k \in \mathbb{S}_i^{IN}} \end{bmatrix} \tag{17}$$

we can rewrite the augmented system (14) compactly as:

$$\begin{aligned} \bar{x}_i^+ &= \bar{A}_i \bar{x}_i + \bar{B}_i u_i + \bar{B}_i^{IN} \bar{u}_i \\ \bar{y}_i &= \bar{C}_i \bar{x}_i \end{aligned} \tag{18}$$

3.2 Optimal control problem and cooperative iterations

We now analyze the global cost function for tracking given in (12), and rewrite it in a way that the specific contribution of each subsystem is highlighted:

$$V_t(\cdot) = \sum_{j=1}^{M} \left(\sum_{k=0}^{N-1} \ell_j(x_j(k)-x_{sj}, u_j(k)-u_{sj}) + V_{fj}(x_j(N)-x_{sj}) \right) + \sum_{j=1}^{M} \|y_{sj}-y_{tj}\|^2_{T_j} \qquad (19)$$

where (x_{sj}, u_{sj}, y_{sj}) represent an equilibrium triple of each subsystem j, y_{tj} is the desired target of the $j-$th subsystem output, and T_j is a positive definite diagonal matrix associated to the output of subsystem j, easily defined from T in (6). We observe that the input of subsystem i only affects the terms associated to the augmented system (14), and hence all terms associated to the other subsystems, i.e. for $j \notin S_i^{OUT}$, can be dropped. More specifically let $\bar{x}_i$ be the current value of the state of the augmented system (18), and $\{\mathbf{u}_j\}_{j\in\mathbb{S}_i^{IN}}$ the finite horizon input sequence of its neighbors; then, the cost function to be minimized by subsystem i reads:

$$\begin{aligned} V_{ti}(\cdot) = \sum_{k=0}^{N-1} & \bar{\ell}_i(\bar{x}_i(k)-\bar{x}_{si}, u_i(k)-u_{si}) \\ & + \bar{V}_{fi}(\bar{x}_i(k)-\bar{x}_{si}) + \|\bar{y}_{si}-\bar{y}_{ti}\|^2_{\bar{T}_i} \quad \text{s.t.} \qquad (20) \\ \bar{x}_i(0) &= \bar{x}_i \\ \bar{x}_i(k+1) &= \bar{A}_i\bar{x}_i(k) + \bar{B}_i u_i(k) + \bar{B}_i^{IN}\bar{u}_i(k) \end{aligned}$$

in which

$$\bar{\ell}_i(\bar{x}_i, u_i) = \frac{1}{2}\left(\bar{x}_i^T\bar{Q}_i\bar{x}_i + u_i^T R_i u_i\right), \quad \bar{V}_{fi}(\bar{x}_i) = \frac{1}{2}\bar{x}_i^T\bar{P}_i\bar{x}_i \qquad (21)$$

Likewise, it is not necessary to include the state constraints of all subsystems, as only those of the augmented system (14) will be affected by the input u_i. Therefore, the proposed method considers the following finite horizon optimal control problem to be solved by each subsystem i:

$$\mathbb{P}_i\left(\bar{x}_i, \{\mathbf{u}_j\}_{j\in\mathbb{S}_i^{IN}}\right): \quad \min_{\mathbf{u}_i, x_s, u_s, y_s} V_{ti}(\mathbf{u}_i, x_s, u_s, y_s)$$

s.t. (5) and

$$\begin{aligned} \mathbf{u}_i \in \bar{\mathscr{U}}_i\left(\bar{x}_i, \{\mathbf{u}_j\}_{j\in\mathbb{S}_i^{IN}}\right) \qquad (22) \\ (\bar{x}_i(N), \bar{y}_{si}) \in \bar{\Omega}_i \end{aligned}$$

in which

$$\bar{\mathscr{U}}_i\left(\bar{x}_i, \{\mathbf{u}_j\}_{j\in\mathbb{S}_i^{IN}}\right) = \{\mathbf{u}_i \mid u_i(k) \in \mathbb{U}_i, \bar{x}_i(k) \in \bar{\mathbb{X}}_i\}$$

with $\bar{\mathbb{X}}_i$ being the state constraint set for the augmented system (18), and $\bar{\Omega}_i$ is an admissible polyhedral invariant set for tracking for system (18), as explained in (Ferramosca et al., 2013). Once we solve problem (22), we obtain an optimal input sequence $\mathbf{u}_i^0$. In similar manner as in cooperative regulation problem, we operate cooperative iterations until a relative error tolerance between the input vector at two consecutive iterations or a maximum number of cooperative iterations are achieved. The detailed calculations are reported formally in Algorithm 1.

Algorithm 1. (Cooperative MPC - Single step). **Require:** Systems (18), $\mathbb{S}_i^{IN}\,\forall i = 1\dots M$, tolerance ε, maximum number of cooperative iterations q_{max}, convex combination weights $w_i > 0$, such that $\sum_{i=1}^{M} w_i = 1$.

```
1:  Set $q \leftarrow 0$ and $e_i \leftarrow 2\varepsilon$.
2:  while $q < q_{max}$ and $\exists i$ such that $e_i > \varepsilon$ do
3:      $q \leftarrow q+1$
4:      for $i = 1$ to $M$ do
5:          Solve problem $\mathbb{P}_i$ in (22) to obtain the optimal input sequence
            $\mathbf{u}_i^0(x)$ and the centralized state-steady triple $(x_s, u_s, y_s)$.
6:          if $q = 1$ then
7:              $\mathbf{u}_i^{[q-1]} = \left[u_{s_i}^T \;\cdots\; u_{s_i}^T\right]^T$
8:          end if
9:          Define new iterate: $\mathbf{u}_i^{[q]} = w_i\mathbf{u}_i^0 + (1-w_i)\mathbf{u}_i^{[q-1]}$.
10:         Compute convergence error: $e_i = \frac{||\mathbf{u}_i^{[q]} - \mathbf{u}_i^{[q-1]}||}{1+||\mathbf{u}_i^{[q]}||}$
11:     end for
12: end while
13: return Overall solution $\mathbf{u} = \left(\mathbf{u}_1^{[q]}, \mathbf{u}_2^{[q]}, \dots, \mathbf{u}_M^{[q]}\right)$.
```

The lines 5-10 of Algorithm 1 are executed (in parallel) by each subsystem defined as in (18). Problem (22) finds, in a single step, the optimal input sequence for subsystem i, $\mathbf{u}_i^0$ and the centralized steady-state triple (u_s, x_s, y_s). It is important to note that subsystems solve their optimization problem independently of each other since there is no communication at this point. Communication takes place after line 10, when each subsystem communicates its local input $\mathbf{u}_i^{[q]}$ and its convergence error e_i.

3.3 A two step variant

A variant of the proposed algorithm is where the steady target problem is executed by each subsystem separately from the optimal control problem. In the first step each subsystem finds the equilibrium triple (x_s, u_s, y_s) for the overall system from (5). Then, each subsystem solves a problem that is similar to $\mathbb{P}_i$ in (22) with (x_s, u_s, y_s) being known parameters instead of decision variables. The problem is solved in parallel by each agent. At the end of each cooperative iteration, communication takes place; each subsystem communicates its local input $\mathbf{u}_i^{[q]}$ and its convergence error e_i.

3.4 Complexity analysis

It is important to remark that in problem (22), each subsystem i computes the evolution trajectory of its augmented system. This is sufficient to minimize the global objective function and enforce state (or output) constraints because the states that are discarded are those that are not affected by the input of subsystem i. On the contrary, general cooperative algorithms keep track of the evolution of the overall state to achieve the same goals. It is well known that Quadratic Programs arising in MPC problems, like (4) or (22), are more effectively solved for large scale systems using Interior Point algorithms with both state and input sequences as decision variables (Rao et al., 1998). Since the augmented system comprises a subset of the overall system state, it follows that proposed single step method has lower (no higher) complexity than the method in (Ferramosca et al., 2013). A summary of the complexity of both methods, as well as of the two step variant discussed in §3.3, is reported in Table 1, in which DMPC0 refers to the method discussed in §2.4, DMPC1 is the proposed single step method discussed in §3.2, and DMPC2 is the proposed two step method discussed in §3.3.

4. APPLICATION EXAMPLE

4.1 Multi-stage evaporator model and subsystems

As an example, we consider a "forward feed", triple effect, evaporator process. The mass and energy balance equations for the three evaporators ($i = 1, 2, 3$) are given by:

$$\begin{cases} \dfrac{dM_i}{dt} = L_{i-1} - L_i - V_i \\ c_p M_i \dfrac{dT_i}{dt} = L_{i-1} c_p \left(T_{i-1} - T_i\right) - V_i \lambda_i + V_{i-1} \lambda_{i-1} \\ M_i \dfrac{d\chi_i}{dt} = L_{i-1} \chi_{i-1} + \left(V_i - L_{i-1}\right) \chi_i \end{cases} \qquad (23)$$

in which the states/outputs are: the liquid mass M_i, the evaporator temperature T_i, the solute mass fraction χ_i; the inputs are: the liquid outlet rate L_i, the vapor outlet rate V_i. For the first evaporator $L_{i-1} = F$ is the feed rate, $T_{i-1} = T_F$ is the feed temperature, $V_{i-1}\lambda_{i-1} = Q_1$ is the external duty. The other parameters are the heat capacity c_p and the heat of evaporation λ_i. Parameters are taken from (Coulson and Richardson, 1993, p.632-633). To obtain a linear model, the nonlinear system (23) is tested for 72 hours in which each input has 2% of maximum amplitude variation. Generalized Binary Noise (GBN) input signals are produced (Zhu, 2001) with a sampling time of 1 min and a a switch probability of 2%. Then, both inputs and outputs are mean centered, and normalized in the range $[-1, 1]$. A MISO approach is considered to identify a linear model for each output, using the N4SID algorithm available in the Systems Identification toolbox in Matlab (Ljung, 1999). A transfer function model for each (nonzero) input-output pair is reported in Table 2, which highlights the sparsity of the identified model. We assume that the process is split into three subsystems: Subsystem 1 has (L_1, Q_1, V_1) as inputs and (M_1, T_1, χ_1) as outputs; Subsystem 2 has (L_2, V_2) as inputs and (M_2, T_2, χ_2) as outputs; Subsystem 3 has (L_3, V_3) as inputs and (M_3, T_3, χ_3) as outputs.

4.2 Controllers, simulations and results

The purpose of this simulation is to steer the outputs of the three subsystems as close as possible to given piece-wise constant set-points. In particular we have the following desired normalized targets: for $k = 0, \dots, 4$ target is the origin; for $k = 5, \dots, 239$ $y_{t1} = [0, 0, 0.07]$, $y_{t2} = [0, 0, 0.1]$, $y_{t3} = [0, 0, 0.7]$; for $k = 240, \dots, 480$ $y_{t1} = [0, 0, -0.07]$, $y_{t2} = [0, 0, -0.1]$, $y_{t3} = [0, 0, -0.7]$. Three controllers are compared: DMPC0 (§2.4); DMPC1(§3.2); DMPC2(§3.3). All controllers are tuned with the same parameters: $N = 100$, $\|u\|_\infty \leq 1$; $Q_i = C_i^T \operatorname{diag}([0.1, 0.1, 1]) C_i$, for $i = 1, 2, 3$; $R_1 = 0.01 I_3$, $R_i = 0.01 I_2$ for $i = 2, 3$; $T_1 = 10^5 I_3$, $T_i = 10^5 \operatorname{diag}([1, 0, 1])$ for $i = 2, 3$; P_i for $i = 1, 2, 3$ is the solution of the discrete algebraic Riccati equation for each subsystem; cooperative loop is iterated until the maximum number of cooperative iteration ($q_{\max} = 100$) are reached or convergence error is less than $\varepsilon = 0.01$. Simulations are performed in Matlab (version 2015b) on a MacBook Pro (3 GHz Intel Core i7, 16 GB RAM). Figure 1 shows the cumulative distribution function (CDF) of computation times. For each time t, the CDF is defined as the fraction of distributed MPC algorithm executions that are solved in time t or less. We notice that DMPC2 completed 97% of its executions in less than 1 second, while DMPC1 completed in the same time about 80% of its executions. On the other hand, DMPC0 completed no executions in 1 second or less; at least it needs about twice computation time (i.e. 2 s) to complete 82% of its executions. From this plot we can appreciate how DMPC1 and DMPC2 are parsimonious with respect to DMPC0. Figure 2 shows the closed-loop evolution of inputs (with constraints) and outputs with relatives desired setpoint. Results are similar for all controllers, and DMPC0 and DMPC1 are (as expected) identical in closed-loop performance. By using augmented state rather than the centralized one, in the prediction model, we have not taken out useful information from the dynamics.

Table 1. Comparison of computational complexity of the three algorithms.

	DMPC0	DMPC1	DMPC2
Prediction model	Centralized	Augmented	Augmented
Target calc. (TC)	Embedded	Embedded	Separate
TC decision var.	–	–	(x_s, u_s, y_s)
OCP decision var.	$(\mathbf{u}_i, \mathbf{x}, x_s, u_s, y_s)$	$(\mathbf{u}_i, \bar{\mathbf{x}}_i, x_s, u_s, y_s)$	$(\mathbf{u}_i, \bar{\mathbf{x}}_i)$

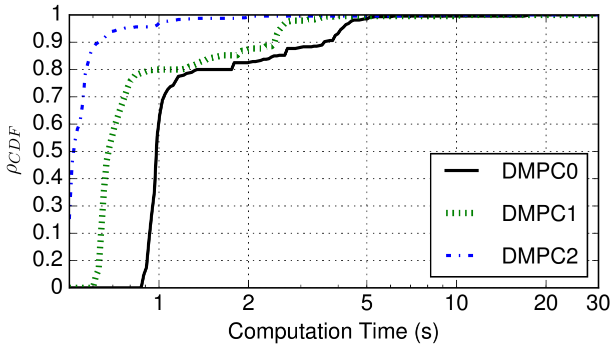

Fig. 1. Cumulative distribution function of computation time.

5. CONCLUSIONS

We presented in this paper cooperative distributed MPC algorithms for tracking piece-wise constant references, in linear systems divided into a finite number of interacting subsystems. The main contribution of this work is to reduce the dimension of the prediction model used by each subsystem, by exploiting basic concepts of graph theory, while retaining the global optimality of the cooperative algorithm. This approach reduces the computational and communication requirements of the proposed algorithm(s) with respect to the currently available ones. Two variants were considered, one with target calculation embedded in the optimal control problem and one with separate target calculation. A multiple effect evaporator process has been presented, and the proposed algorithms were shown to be significantly more effective in terms of average and worst-case computation time.

REFERENCES

Bemporad, A. and Barcelli, D. (2010). Decentralized model predictive control. In A. Bemporad, M. Heemels, and M. Johansson (eds.), *Networked Control Systems*, Springer Lecture Notes in Control and Information Sciences. Springer.

Betti, G., Farina, M., and Scattolini, R. (2014). Distributed MPC: A noncooperative approach based on robustness concepts. In J.M. Maestre and R.R. Negenborn (eds.), *Distributed Model Predictive Control Made Easy*, chapter 26. Springer, Upper Saddle River, New Jersey.

Coulson, J.M. and Richardson, J.F. (1993). *Chemical Engineering*, volume 2. Pergamon Press.

Farina, M. and Scattolini, R. (2012). Distributed predictive control: A non-cooperative algorithm with neighbor-to-neighbor communication for linear systems. *Automatica*, 48, 1088–1096.

Ferramosca, A., Limon, D., Alvarado, I., and Camacho, E.F. (2013). Cooperative distributed MPC for tracking. *Automatica*, 49, 906–914.

Table 2. Triple effect evaporator model in transfer functions (normalized inputs and outputs)

	L_1	Q_1	V_1	L_2	V_2	L_3	V_3
M_1	$-\frac{0.04797}{z-2.717}$	-	$-\frac{0.0339}{z-2.717}$	-	-	-	-
T_1	-	$\frac{0.564}{z-2.509}$	$-\frac{0.1745}{z-2.509}$	-	-	-	-
χ_1	-	-	$\frac{0.009394}{z-2.549}$	-	-	-	-
M_2	$\frac{0.05726}{z-2.716}$	-	-	$-\frac{0.07207}{z-2.716}$	$-\frac{0.09465}{z-2.716}$	-	-
T_2	$\frac{0.008029z-0.01856}{z^2-4.913z+6.029}$	$\frac{0.089z-0.02579}{z^2-4.913z+6.029}$	$\frac{0.2431z-0.6396}{z^2-4.913z+6.029}$	-	$\frac{-0.6057z+1.451}{z^2-4.913z+6.029}$	-	-
χ_2	$-\frac{0.01418}{z-2.604}$	-	$\frac{0.01038}{z-2.604}$	-	$\frac{0.02976}{z-2.604}$	-	-
M_3	-	-	-	$\frac{0.07503}{z-2.712}$	-	$-\frac{0.08504}{z-2.712}$	$-\frac{0.1255}{z-2.712}$
T_3	$\frac{0.001138z+0.03875}{z^2-4.898z+5.986}$	$\frac{-0.02526z+0.3423}{z^2-4.898z+5.986}$	$\frac{0.06671z-0.127}{z^2-4.898z+5.986}$	$\frac{0.09903z-0.2557}{z^2-4.898z+5.986}$	$\frac{2.472z-6.521}{z^2-4.898z+5.986}$	-	$\frac{-2.895z+7.385}{z^2-4.898z+5.986}$
χ_3	$\frac{-0.01013z+0.01865}{z^2-5.241z+6.864}$	-	$\frac{0.004064z-0.005355}{z^2-5.241z+6.864}$	$\frac{-0.2224z+0.6029}{z^2-5.241z+6.864}$	$\frac{0.01244z-0.02893}{z^2-5.241z+6.864}$	-	$\frac{0.464z-1.249}{z^2-5.241z+6.864}$

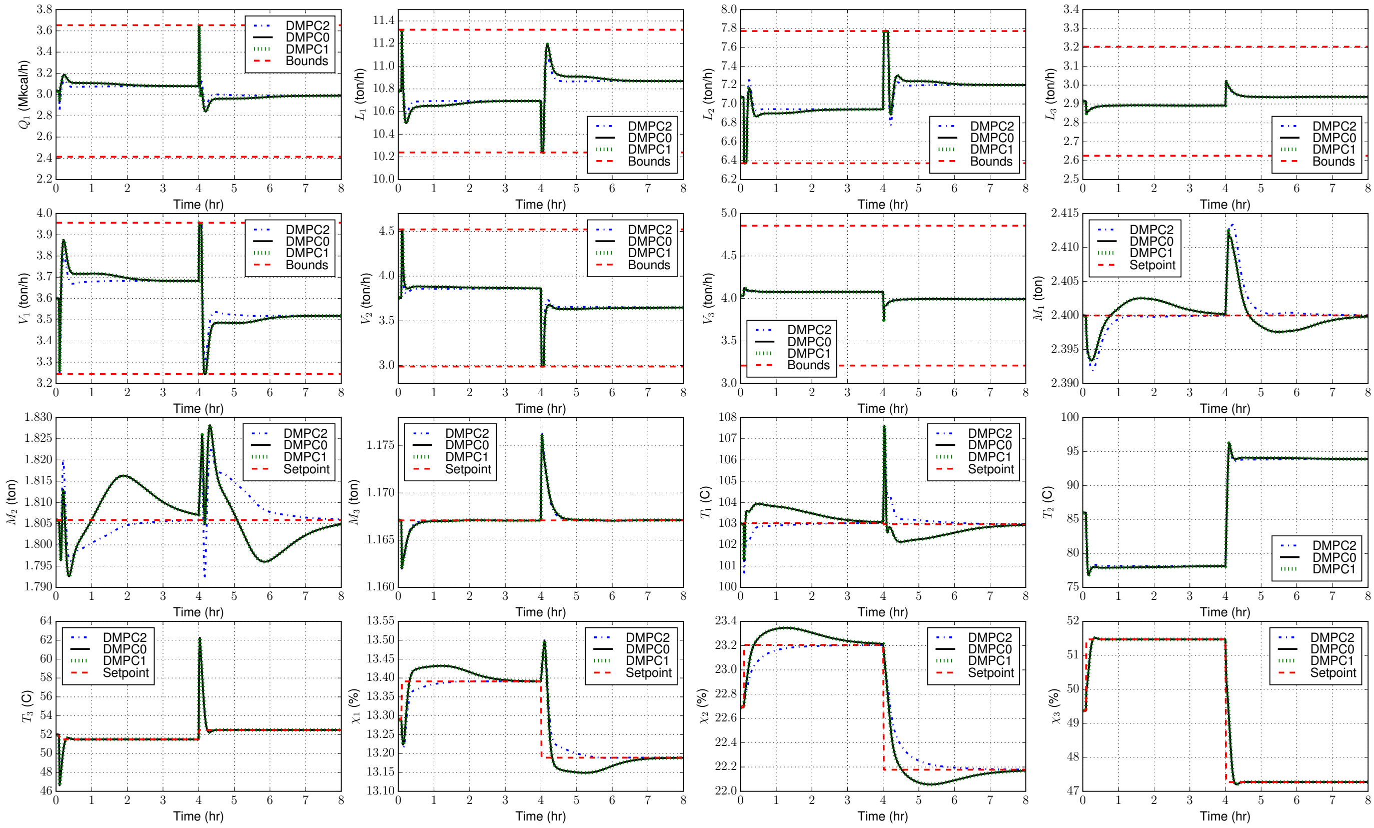

Fig. 2. Closed-loop evolution of inputs and outputs. Red dashed lines are used for input bounds or output setpoints.

Limon, D., Alvarado, I., Alamo, T., and Camacho, E. (2008). MPC for tracking piecewise constant references for constrained linear systems. *Automatica*, 44, 2382–2387.

Ljung, L. (1999). *System Identification: Theory for the User*. Prentice Hall Inc., Upper Saddle River, New Jersey, second edition.

Maciejowski, M.J. (2000). *Predictive Control with Constraints*. Prentice Hall.

Pannocchia, G. (2014). Distributed model predictive control. In B. T. Samad (ed.), Encyclopedia of Systems and Control, chapter 5, 302–308. Springer.

Pannocchia, G., Rawlings, J.B., and Wright, S.J. (2011). Conditions under which suboptimal nonlinear MPC is inherently robust. *Syst. Contr. Lett.*, 60, 747–755.

Rao, C.V., Wright, S.J., and Rawlings, J.B. (1998). Application of interior-point methods to model predictive control. *J. Optim. Theory Appl.*, 99, 723–757.

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

Riverso, S., Farina, M., and Ferrari-Trecate, G. (2013). Plug-and-play decentralized model predictive control for linear systems. *IEEE Trans. Auto. Contr.*, 2608–2614.

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

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

Zhu, Y. (2001). *Multivariable System Identification for Process Control*. Springer.