

Parsimonious Cooperative Distributed MPC Algorithms for Offset-free Tracking [☆]

Matteo Razzanelli¹

University of Pisa – Department of Information Engineering. Via G. Caruso 15, Pisa, Italy

Gabriele Pannocchia²

University of Pisa – Department of Industrial and Civil Engineering. Largo Lucio Lazzarino 2, Pisa, Italy

Abstract

We propose in this paper novel cooperative distributed MPC algorithms for tracking of piecewise constant setpoints in linear discrete-time systems. The available literature for cooperative tracking requires that each local controller uses the centralized state dynamics while optimizing over its local input sequence. Furthermore, each local controller must consider a centralized target model. The proposed algorithms instead use a suitably augmented local system, which in general has lower dimension compared to the centralized system. The same parsimonious parameterization is exploited to define a target model in which only a subset of the overall steady-state input is the decision variable. Consequently the optimization problems to be solved by each local controller are made simpler. We also present a distributed offset-free MPC algorithm for tracking in the presence of modeling errors and disturbances, and we illustrate the main features and advantages of the proposed methods by means of a multiple evaporator process case study.

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

1. Introduction

Model predictive control (MPC) is a class of advanced control techniques, widely used especially in the process industries, able to ensure closed-loop stability, constraint satisfaction and robustness for multivariable linear and nonlinear systems [2, 3, 4]. Typical theoretical results on MPC consider the regulation problem, i.e. steering the system state to the origin. In general, we can take into account a non-zero setpoint by shifting the origin of the system. Depending on the extent of the required setpoint change, feasibility issues may arise. This problem can be avoided, as explained in [5] where a centralized MPC for tracking a constrained linear systems is proposed, which ensures nominal recursive feasibility and stability of the closed-loop system under any change of the setpoint and tractable computation for linear and non-linear systems [2, 6, 7].

Large-scale systems with several interconnected units may exchange material, energy and information streams. The global performance of these systems (e.g. industrial processing plants, power generation networks, etc.) depends on local profitability and on the level of interaction among subsystems. Optimal global performance could be obtained using a single centralized MPC algorithm for all interconnected units. However, this choice is often avoided for several reasons. When the overall

number of inputs and states is very large, a single optimization problem may require computational resources (CPU time, memory, etc.) that are not available and/or compatible with system's dynamics. We may say that problem is not scalable. Even if this limitation does not hold, it is often the case that organizational reasons impose the use of smaller local controllers, which are easier to maintain and operate. **For instance, the process to be controlled may be spread geographically. By using a centralized approach, when a partial shutdown on the network happens, the rest of the units would stop working.**

Thus, industrial control systems are often *decentralized*, i.e. the overall system is divided into subsystems and a local controller is designed for each unit disregarding interactions from/to other subsystems. Due to dynamic coupling it is well known that performance of such decentralized systems may be poor, and stability properties may be even lost unless suitable precautions are employed [8, 9]. Between centralized and decentralized strategies [8, 9], *distributed* control algorithms preserve topology and flexibility of decentralized controllers and may offer nominal closed-loop stability guarantees. This is achieved by two features: the interactions among subsystems are explicitly modeled and open-loop information, usually input trajectories, is exchanged among subsystem controllers [10]. **In [11] a framework to avoid the loss of feasibility, guarantee stability, constraint satisfaction as well as convergence to admissible set-points is proposed in a cooperative fashion.** Moreover, in distributed control there are two main strategies for utilization of the open-loop information. In *non-cooperative* distributed control, each subsystem controller anticipates the effect of in-

[☆]A preliminary version of this work was presented as Keynote lecture at the IFAC Symposium DYCOPS-CAB 2016 [1].

¹e-mail: matteo.razzanelli@ing.unipi.it

²e-mail: gabriele.pannocchia@unipi.it. Corresponding Author.

interactions only locally, i.e. to optimize a local objective function [12, 13]. 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 [3, 14] 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 [15] 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 [15] to develop novel cooperative distributed MPC approaches for tracking that rely on information that is “as local as possible”, and to include offset-free action in spite of modeling errors.

The rest of this paper is organized as follows. In Section 2 preliminaries on centralized and distributed MPC, for regulation and tracking, are described. In Section 3 we present and discuss new distributed MPC formulations for tracking. In Section 4 we extend the proposed methods to achieve offset-free tracking in the presence of disturbances or plant/model mismatch. In Section 5 we present a numerical example to show the benefits of the proposed methods. Then, in Section 6 the main conclusions of this work are summarized and open problems are sketched.

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 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 $\|x\|_Q^2 = x^T Q x$. The superscript 0 denotes an optimal cost or vector. Operators $\text{diag}\{T_1, \dots, T_M\}$ and $\text{hor}\{T_1, \dots, T_M\}$ represent block diagonal and block horizontal concatenation, respectively, of the supplied matrices $\{T_1, \dots, T_M\}$, assumed to have suitable dimensions.

2. Preliminary definitions and related work

2.1. Centralized system and local subsystems

We focus in this paper on discrete-time, linear, time-invariant systems (DLTI) in the form:

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

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. We assume that the overall system (1) can be represented as the union of M DLTI subsystems. Each subsystem i , with $i \in \{1, \dots, M\}$, has local input $u_i \in \mathbb{R}^{m_i}$ and local output $y_i \in \mathbb{R}^{p_i}$, and its evolution is given by:

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

where \mathcal{N}_i is the set of neighbors of subsystem i , and $x_i \in \mathbb{R}^{n_i}$, $x_i^+ \in \mathbb{R}^{n_i}$ are the local state and successor state.

We remark that, without loss of generality, in (2) it is not necessary to consider state interactions among subsystems, because the “local” state x_i may be augmented (if necessary) to include other required state interaction terms.

We make the following assumption throughout the paper, except in Section 4 where the state measurement requirement is relaxed.

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

$$\text{rank} \begin{bmatrix} A_i - I & B_i \\ C_i & 0 \end{bmatrix} = n_i + p_i \quad (3)$$

We remark that Assumption 1 implies that (A, B) is controllable and the condition

$$\text{rank} \begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix} = n + p \quad (4)$$

holds true.

2.2. Useful graph theory reminders

We recall a few concepts from graph theory, in order to represent and analyze the different subsystems.

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is composed by a finite set of vertices (or nodes) \mathcal{V} and a set of edges (or lines) $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ that connect pairs of vertices. When edges are not oriented, we call this type of graph *undirected*. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *directed* if \mathcal{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 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the *inlet star* and the *outlet star* of node v_i are, respectively, the following sets:

$$S_i^{IN} = \{v_j \in \mathcal{V} \mid (v_j, v_i) \in \mathcal{E}\}, S_i^{OUT} = \{v_j \in \mathcal{V} \mid (v_i, v_j) \in \mathcal{E}\}$$

We can simplify the notation, by indicating with i the generic node v_i , and with j the generic node v_j belonging to inlet star of i , ($j \in S_i^{IN}$), or to outlet star of i , i.e. ($j \in S_i^{OUT}$). Then, 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 ($j \in S_i^{IN}$), whereas S_i^{OUT} is the set of subsystems of which subsystem i is neighbor.

As an illustrative example, consider the following DLTI system:

$$\begin{aligned} x_1^+ &= A_1 x_1 + B_1 u_1 \\ x_2^+ &= A_2 x_2 + B_2 u_2 + B_{21} u_1 \\ x_3^+ &= A_3 x_3 + B_3 u_3 + B_{31} u_1 + B_{32} u_2 \end{aligned} \quad (5)$$

Node 1 has an empty inlet star set, so $\mathcal{N}_1 = \emptyset$. Node 2 is influenced by node 1, i.e. $\mathcal{N}_2 = \{1\}$, while node 3 is influenced by node 1 and node 2. i.e. $\mathcal{N}_3 = \{1, 2\}$. The system (5) is associated with the directed graph shown in Fig. 1. Moreover, the

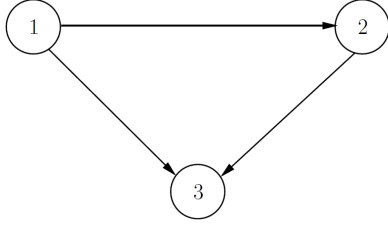


Figure 1: Directed graph corresponding to system (5).

corresponding overall system in form (1) is given by:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}^+ = \begin{bmatrix} A_1 & 0 & 0 \\ 0 & A_2 & 0 \\ 0 & 0 & A_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} B_1 & 0 & 0 \\ B_{21} & B_2 & 0 \\ B_{31} & B_{32} & B_3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

2.3. Brief review of centralized MPC

In order to set the bases of the proposed methods, we first recall the standard formulation of centralized MPC.

2.3.1. Centralized regulation

Consider the centralized DLTI system (1), note that $(x, u) = (0, 0)$ is an equilibrium point, and let N be a positive horizon length. We consider the problem of steering a given initial state to the equilibrium while respecting input and state constraints:

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

in which $\mathbb{X} \subset \mathbb{R}^n$ and $\mathbb{U} \subset \mathbb{R}^m$ are compact polyhedral sets containing the origin. Given the current state, x , and a finite-horizon input sequence $\mathbf{u} = \{u(0), \dots, u(N-1)\}$ we define the cost function over the finite horizon as follows:

$$\begin{aligned} V(x, \mathbf{u}) &= \sum_{k=0}^{N-1} \ell(x(k), u(k)) + V_f(x(N)) \quad \text{s.t.} \\ x(0) &= x \\ x(k+1) &= Ax(k) + Bu(k) \quad k = 0, \dots, N-1 \end{aligned} \quad (6)$$

in which $\ell(\cdot)$ is the stage cost function given by:

$$\ell(x, u) = \frac{1}{2} (x^T Q x + u^T R u)$$

with Q, R positive definite matrices, and $V_f(\cdot)$ is the terminal cost function given by:

$$V_f(x) = \frac{1}{2} x^T P x \quad (7)$$

in which P is also positive definite.

We can now define the finite horizon optimal control problem (FHOCP) to be solved given the current state x as follows:

$$\mathbb{P}(x) : \quad \min_{\mathbf{u}} \{V(x, \mathbf{u}) \mid \mathbf{u} \in \mathcal{U}_N(x)\} \quad (8)$$

where:

$$\mathcal{U}_N(x) = \{\mathbf{u} \mid u(k) \in \mathbb{U}, x(k) \in \mathbb{X}, x(N) \in \mathbb{X}_f \subseteq \mathbb{X}\}$$

Solving problem $\mathbb{P}(x)$ in (8) provides $\mathbf{u}^0(x)$ as optimal input sequence, associated with a corresponding optimal state sequence $\mathbf{x}^0(x) = \{x^0(0) = x, x^0(1), \dots, x^0(N)\}$. Then, only the first element of the optimal input sequence is sent to the controlled system, i.e. $u = u^0(0)$. Finally, we recall that $\mathbb{P}(x)$ can be posed as a Quadratic Program (QP) and effectively solved numerically.

2.3.2. Centralized tracking

In general, it is a common objective to drive the output 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 equilibrium relation must hold:

$$\begin{bmatrix} A-I & B & 0 \\ C & 0 & -I \end{bmatrix} \begin{bmatrix} x_s \\ u_s \\ y_s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (9)$$

Notice that for any $y_s \in \mathbb{R}^p$, condition (4) implies that there exists a (non necessarily unique) pair (x_s, u_s) such that (9) holds true. We can now define the output target offset cost as:

$$V_{ss}(y_s, y_t) = \|y_s - y_t\|_T^2 \quad (10)$$

with T positive definite diagonal matrix. Then, 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:

$$\min_{x_s, u_s, y_s} V_{ss}(y_s, y_t) \quad \text{s.t. (9) and } x_s \in \mathbb{X}, \quad u_s \in \mathbb{U} \quad (11)$$

The above is often referred to as a steady-state target optimizer (SSTO) problem, and we remark that (11) is also a QP.

Given (x_s^0, u_s^0, y_s^0) , solution to problem (11), 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:

$$\mathbb{P}(\tilde{x}) : \quad \min_{\tilde{\mathbf{u}}} \{V(\tilde{x}(0), \tilde{\mathbf{u}}) \mid \tilde{\mathbf{u}} \in \tilde{\mathcal{U}}_N(\tilde{x})\} \quad (12)$$

where:

$$\tilde{\mathcal{U}}_N(\tilde{x}) = \{\tilde{\mathbf{u}} \mid \tilde{u}(k) + u_s^0 \in \mathbb{U}, \tilde{x}(k) + x_s^0 \in \mathbb{X}, \tilde{x}(N) \in \mathbb{X}_f\}$$

The receding horizon control law uses only the first element of the optimal sequence $\tilde{\mathbf{u}}^0(x)$ solution to (12), so the input will be $u = \tilde{u}^0(0) + u_s^0$. Notice that in some tracking formulations [5] the steady-state problem (11) can be embedded into problem $\mathbb{P}(\tilde{x})$, resulting in a single-layer MPC structure. **The reason behind this approach is that an external steady-state optimization problem could generate a state target x_s such that the corresponding terminal region may be not reachable by N steps from the the current initial state $x(0)$ while respecting input and state constraints [5].**

2.4. Brief review of distributed MPC

We now review the basic cooperative distributed MPC algorithms, for regulation first and then for tracking.

2.4.1. Cooperative distributed regulation

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

$$\begin{aligned} V_i(x_i, \mathbf{u}_i, \{\mathbf{u}_j\}_{j \in \mathcal{N}_i}) &= \sum_{k=0}^{N-1} \ell_i(x_i(k), u_i(k)) + V_{f_i}(x_i(N)) \\ \text{s.t. } x_i(0) &= x_i \\ x_i(k+1) &= A_i x_i(k) + B_i u_i(k) + \sum_{j \in \mathcal{N}_i} B_{ij} u_j(k) \end{aligned} \quad (13)$$

in which

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

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 [3, 10, 16], each subsystem optimizes the same global objective function, defined as

$$V(x, \mathbf{u}) = \sum_{i=1}^M V_i(x_i, \mathbf{u}_i, \{\mathbf{u}_j\}_{j \neq i}) \quad (14)$$

We notice that the global cost function (14) is equivalent to that of the centralized problem (6) with the following weight matrices

$$\begin{aligned} Q &= \text{diag}\{Q_1, \dots, Q_M\}, \quad R = \text{diag}\{R_1, \dots, R_M\}, \\ P &= \text{diag}\{P_1, \dots, P_M\} \end{aligned}$$

Hence, the FHOCP to be solved by each subsystem i reads:

$$\begin{aligned} \mathbb{P}_i(x, \{\mathbf{u}_j\}_{j \neq i}) : \quad \min_{\mathbf{u}_i} V(x, \mathbf{u}) \quad \text{s.t.} \\ \mathbf{u}_i \in \mathcal{U}_i(x, \{\mathbf{u}_j\}_{j \neq i}) \\ x(N) \in \mathbb{X}_f \subseteq \mathbb{X} \end{aligned} \quad (15)$$

in which

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

We remark that each subsystem solves problem (15) 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.

At the q -th cooperative iteration ($q = 1$, at the beginning of this iterative process), let $\mathbf{u}_i^{[q-1]}$ be the known value of subsystem i input sequence. We solve problem $\mathbb{P}_i(x, \{\mathbf{u}_j^{[q-1]}\}_{j \neq i})$ in (15) to obtain \mathbf{u}_i^0 . 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.:

$$\mathbf{u}_i^{[q]} = w_i \mathbf{u}_i^0 + (1 - w_i) \mathbf{u}_i^{[q-1]} \quad (16)$$

in which $w_i > 0$ such that $\sum_{i=1}^M w_i = 1$. Cooperative iterations are typically performed until convergence, i.e. when $\|\mathbf{u}_i^{[q]} - \mathbf{u}_i^{[q-1]}\|$ 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. We remark that nominal stability is guaranteed for any finite number of cooperative iterations [3, 16].

2.4.2. Cooperative distributed tracking

Problem (15) described above can be applied to tracking or can be integrated with dynamic optimization layer as in [15]. Specifically, an *artificial* equilibrium triple (x_s, u_s, y_s) of the overall centralized system (1), i.e. satisfying (9), is added as decision variable. Consequently, the global cost function is modified by adding a target offset term as in (57a) that penalizes the deviation of y_s from y_t , to obtain:

$$\begin{aligned} V_i(x, \mathbf{u}, x_s, u_s, y_s) &= V_{ss}(y_s, y_t) + \sum_{k=0}^{N-1} \ell(x(k) - x_s, u(k) - u_s) + \\ &\quad V_f(x(N) - x_s) \\ \text{s.t. } x(0) &= x \\ x(k+1) &= Ax(k) + Bu(k) \end{aligned} \quad (17)$$

Then, the FHOCP to be solved by each subsystem i reads:

$$\mathbb{P}_i(x, \{\mathbf{u}_j\}_{j \neq i}) : \quad \min_{\mathbf{u}_i, x_s, u_s, y_s} V_i(x, \mathbf{u}, x_s, u_s, y_s) \quad (18a)$$

s.t. (9) and

$$\mathbf{u}_i \in \mathcal{U}_i(x, \{\mathbf{u}_j\}_{j \neq i}) \quad (18a)$$

$$(x(N), y_s) \in \Omega \quad (18a)$$

where Ω is an admissible polyhedral invariant set for tracking for system (1), as explained in [15]. We also point out that in problem (18) solved by [15], the centralized target triple (x_s, u_s, y_s) is parameterized by y_s only, which represents together with \mathbf{u}_i the actual decision variable of each local controller.

3. Proposed methods for nominal tracking

In this section we propose a new cooperative distributed MPC algorithm for nominal tracking, and then we present several variants. Differently from the available approaches used in cooperative MPC algorithms [3, 15, 16], in the proposed methods each local controller does not have to keep track the centralized system state but only of a subset of this one that is strictly necessary to achieve the desired global tracking goal. In this

way two main results can be accomplished. Firstly, the communication among subsystems is reduced, and secondly the optimization problem solved by each local controller is made simpler (reduced in dimension), and hence computation time may be reduced.

The first proposed method is based on a single optimization layer as described in § 2.4, and then, we present a two step variant algorithm. In these two methods, the equilibrium target is still in centralized form, but a further variant is then described, which uses an augmented target model.

3.1. Defining the augmented system

We analyze each subsystem i , and observe that:

- this subsystem is influenced by the inputs of its *inlet star*, as described in (2);
- the input of this subsystem influences the subsystems of its *outlet star*, along with the inputs of their inlet stars.

The evolution of its own state x_i and of the states of subsystems belonging to its *outlet star* are given by:

$$x_i^+ = A_i x_i + B_i u_i + \sum_{k \in S_i^{IN}} B_{ik} u_k \quad (19)$$

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

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

$$x_j^+ = A_j x_j + \left(B_j u_j + \sum_{k \in S_j^{IN}} B_{jk} u_k \right), \quad j \notin S_i^{OUT} \quad (20)$$

It is therefore clear that each subsystem i should only consider the evolution of subsystems reported in (19), because those reported in (20) are independent of u_i .

We now refer to (19) as the ‘‘augmented system’’. In order to write it in a more convenient way, we can define a new set representing its inlet star:

$$\mathbb{S}_i^{IN} = S_i^{IN} \cup S_i^{OUT} \cup \left(\bigcup_{j \in S_i^{OUT}} S_j^{IN} \setminus \{i\} \right) \quad (21)$$

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

$$\bar{x}_i = \begin{bmatrix} x_i \\ [x_j]_{j \in S_i^{OUT}} \end{bmatrix}, \quad \bar{u}_i = [u_k]_{k \in \mathbb{S}_i^{IN}}, \quad \bar{y}_i = \begin{bmatrix} y_i \\ [y_j]_{j \in S_i^{OUT}} \end{bmatrix} \quad (22)$$

and the following matrices:

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

$$\bar{B}_i^{IN} = \begin{bmatrix} \text{hor}\{B_{ik}\}_{k \in \mathbb{S}_i^{IN}} \\ \text{hor}\{B_{jk}\}_{j \in S_i^{OUT}, k \in \mathbb{S}_i^{IN}} \end{bmatrix} \quad (23)$$

Based on these definitions we can rewrite the augmented system (19) in a more compact form:

$$\begin{aligned} \bar{x}_i^+ &= \bar{A}_i \bar{x}_i + \bar{B}_i u_i + \bar{B}_i^{IN} \bar{u}_i \\ \bar{y}_i &= \bar{C}_i \bar{x}_i \end{aligned} \quad (24)$$

3.2. A single step MPC algorithm based on the augmented system

We now analyze the global cost function for tracking given in (17), and rewrite it in a way that the specific contribution of each subsystem is highlighted:

$$V_i(\cdot) = V_{ss}(y_s, y_t) + \sum_{k=0}^{N-1} \sum_{j=1}^M \ell_j(x_j(k) - x_{sj}, u_j(k) - u_{sj}) + \sum_{j=1}^M V_{fj}(x_j(N) - x_{sj}) \quad (25)$$

where (x_{sj}, u_{sj}) represent state and input equilibrium values of each subsystem j . We observe that the input of subsystem i only affects the terms associated to the augmented system (19), 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 (24), define \bar{x}_{si} as the appropriate selection of components of the centralized state target x_s , and let $\{\mathbf{u}_j\}_{j \in \mathbb{S}_i^{IN}}$ be a finite horizon input sequence of neighbors of the augmented system i . Then, the cost function to be minimized by subsystem i reads:

$$\begin{aligned} V_{ii}(\cdot) &= \sum_{k=0}^{N-1} \bar{\ell}_i(\bar{x}_i(k) - \bar{x}_{si}, u_i(k) - u_{si}) \\ &\quad + \bar{V}_{fi}(\bar{x}_i(k) - \bar{x}_{si}) + V_{ss}(y_s, y_t) \quad \text{s.t.} \quad (26) \\ \bar{x}_i(0) &= \bar{x}_i \\ \bar{x}_i(k+1) &= \bar{A}_i \bar{x}_i(k) + \bar{B}_i u_i(k) + \bar{B}_i^{IN} \bar{u}_i(k) \end{aligned}$$

in which

$$\bar{\ell}_i(\bar{x}_i, u_i) = \frac{1}{2} (\bar{x}_i^T \bar{Q}_i \bar{x}_i + u_i^T R_i u_i), \quad \bar{V}_{fi}(\bar{x}_i) = \frac{1}{2} \bar{x}_i^T \bar{P}_i \bar{x}_i \quad (27)$$

with

$$\bar{Q}_i = \text{diag} \left\{ Q_i, \{Q_j\}_{j \in S_i^{OUT}} \right\}, \quad \bar{P}_i = \text{diag} \left\{ P_i, \{P_j\}_{j \in S_i^{OUT}} \right\} \quad (28)$$

Likewise, it is not necessary to include the state constraints of all subsystems, as only those of the augmented system (19) will be affected by the input u_i . Thus, the proposed method considers the following FHOCP to be solved by each subsystem i :

$$\mathbb{P}_i(\bar{x}_i, \{\mathbf{u}_j\}_{j \in \mathbb{S}_i^{IN}}) : \min_{\mathbf{u}_i, x_s, u_s, y_s} V_{ii}(\mathbf{u}_i, x_s, u_s, y_s) \quad (29a)$$

s.t. (9) and

$$\mathbf{u}_i \in \bar{\mathcal{U}}_i(\bar{x}_i, \{\mathbf{u}_j\}_{j \in \mathbb{S}_i^{IN}}) \quad (29b)$$

$$(\bar{x}_i(N), \bar{y}_i) \in \bar{\Omega}_i \quad (29c)$$

in which

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

where $\bar{\mathbb{X}}_i$ is the state constraint set for the augmented system (24), \bar{y}_{s_i} is the appropriate selection of components of the centralized output target y_s , and $\bar{\Omega}_i$ is an admissible polyhedral invariant set for tracking for system (24), as explained in [15]. Once we solve problem (29), we obtain an optimal input sequence \mathbf{u}_i^0 . Then, we can perform 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 (24), $\mathbb{S}_i^N \forall i = 1 \dots M$, tolerance ε , max. no. coop. iterations q_{\max} , convex combination weights $w_i > 0$, s.t. $\sum_{i=1}^M w_i = 1$.

- 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 (29) 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: $\mathbf{u}_i^{[q-1]} = [u_{s_i}^T \dots u_{s_i}^T]^T$
 - 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 $\mathbf{u} = (\mathbf{u}_1^{[q]}, \mathbf{u}_2^{[q]}, \dots, \mathbf{u}_M^{[q]})$.
-

We remark that lines 5-10 of Algorithm 1 are executed in parallel by each subsystem. Problem (29) 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 .

Remark 2. Line 5 of Algorithm 1 shows that FHOCP (29) computes an optimal input control sequence \mathbf{u}_i^0 . This vector must belong to a set $\bar{\mathcal{U}}_i$ given by:

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

This means that equation (19), implicitly recalled in $\bar{\mathcal{U}}_i$, contains all the state dynamics required to solve the optimal control problem. On the other hand, all those state dynamics in (20) are independent from \mathbf{u}_i so they do appear in $\bar{\mathcal{U}}_i$. The optimal control problem solved in [15] differs from our approach because

plant-wide state dynamics are considered in the FHOCP, i.e. using both equations (19) and (20). This means that FHOCP (29) and the one given in [15], i.e. FHOCP (18), obtain the same solution \mathbf{u}_i^0 .

Proposition 3. For any initial condition $x(0)$ and any given setpoint y_t , the closed-loop system (1) under the receding horizon control law obtained by Algorithm 1 is asymptotically stable and converges to an equilibrium (x_s^*, u_s^*, y_s^*) that is solution to:

$$\mathbb{P}_{ss}(y_t) : \min_{(x_s, u_s, y_s)} V_{ss}(y_s, y_t) \quad s.t. \quad (30a)$$

$$x_s = Ax_s + Bu_s \quad (30b)$$

$$y_s = Cx_s \quad (30c)$$

$$x_s \in \mathbb{X}, \quad u_s \in \mathbb{U} \quad (30d)$$

PROOF. The proof follows from Th. 1 in [15] and Remark 2.

Remark 4. The equilibrium computed in (30) is the best achievable target, i.e. the target in which the weighted distance between the actual target y_s^* and the setpoint y_t is minimized. If the setpoint is actually reachable while respecting input and state constraints, then $y_s^* = y_t$ and $V_{ss}(y_s^*, y_t) = 0$.

To better understand our method we introduce a block diagram in Figure 2 representing step by step Algorithm 1, as well as covering Algorithm 2 discussed in the next paragraph.

3.3. A two step variant

A variant of the proposed algorithm is presented, in which 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^*) of the overall centralized system from (30). Then, each subsystem solves an FHOCP that is similar to \mathbb{P}_i in (29) with $(x_s, u_s, y_s) = (x_s^*, u_s^*, y_s^*)$ being known parameters instead of decision variables. 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 . The detailed calculations are reported formally in Algorithm 2.

Algorithm 2 (Cooperative MPC - Two steps). Require:

Systems (24), $\mathbb{S}_i^N \forall i = 1 \dots M$, tolerance ε , max. no. coop. iterations q_{\max} , convex combination weights $w_i > 0$, s.t. $\sum_{i=1}^M w_i = 1$.

- 1: Solve (30) to obtain the centralized steady-state triple (x_s^*, u_s^*, y_s^*) .
- 2: Set $q \leftarrow 1$ and $e_i \leftarrow 2\varepsilon$.
- 3: **while** $q < q_{\max}$ and $\exists i$ such that $e_i > \varepsilon$ **do**
- 4: $q \leftarrow q + 1$
- 5: **for** $i = 1$ **to** M **do**
- 6: **if** $q = 1$ **then**
- 7: $\mathbf{u}_i^{[q-1]} = [u_{s_i}^T \dots u_{s_i}^T]^T$

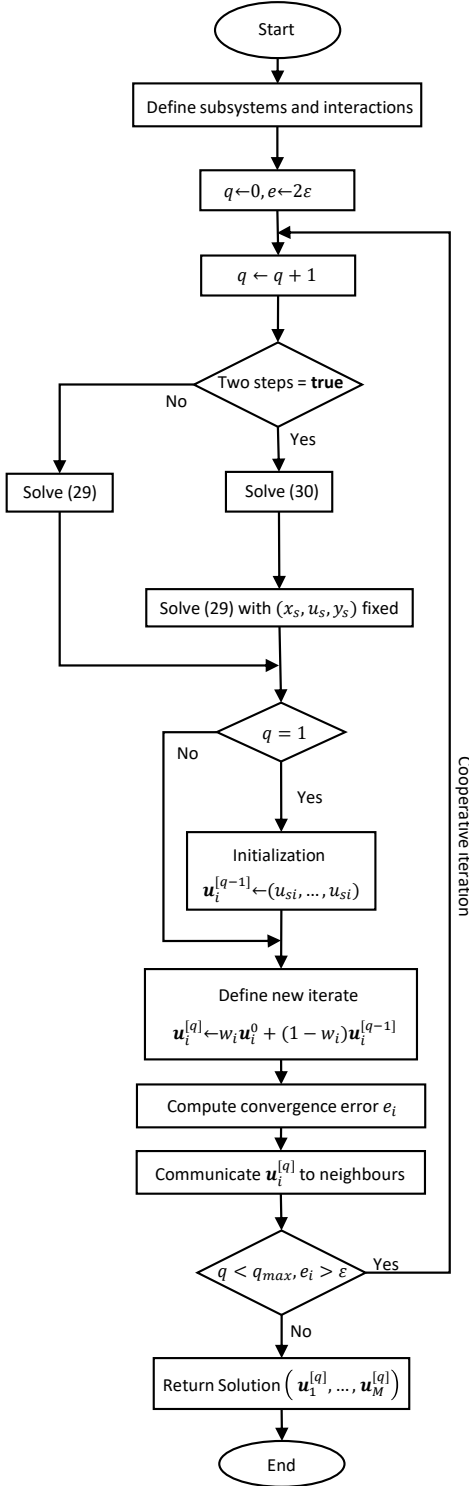


Figure 2: Block diagram of Algorithm 1 and Algorithm 2.

- 8: **end if**
- 9: Solve problem \mathbb{P}_i in (29) with $(x_s, u_s, y_s) = (x_s^*, u_s^*, y_s^*)$ to obtain the optimal input sequence $\mathbf{u}_i^0(x)$.
- 10: Define new iterate: $\mathbf{u}_i^{[q]} = w_i \mathbf{u}_i^0 + (1 - w_i) \mathbf{u}_i^{[q-1]}$.
- 11: Compute convergence error: $e_i = \frac{\|\mathbf{u}_i^{[q]} - \mathbf{u}_i^{[q-1]}\|}{1 + \|\mathbf{u}_i^{[q]}\|}$
- 12: **end for**
- 13: **end while**
- 14: **return** Overall solution $\mathbf{u} = (\mathbf{u}_1^{[q]}, \mathbf{u}_2^{[q]}, \dots, \mathbf{u}_M^{[q]})$.

It is important to observe that in Algorithm 2, line 1 is solved by each subsystem. Alternatively, it could be solved by a coordinator, but this would diminish the distributed architecture. Moreover, similarly to Algorithm 1, lines 5-11 of Algorithm 2 are executed in parallel by each subsystem, and communication takes place after line 11, when each subsystem communicates its local input $\mathbf{u}_i^{[q]}$ and its convergence error e_i .

Remark 5. Note that the equilibrium triple (u_s, x_s, y_s) in Algorithm 2 is computed externally of the cooperative loop and of the FHOCP as described at Line 9. Hence, the final solution \mathbf{u} computed at Line 14 of Algorithm 2 may differ from the other one computed in the Algorithm 1 at Line 13.

3.4. Using an augmented target model

From (29) we observe again that the equilibrium triple is still in centralized form, so the distributed architecture is still not exploited at maximum. In order to distribute the target equilibrium computation, we would like to make use of the augmented system (24). However, we cannot use only u_{si} as decision variable to minimize the offset cost of all outputs \bar{y}_{si} in the augmented system (24). To overcome this limitation, we rewrite the augmented system (24) in an equivalent form that considers as target decision variables the local inputs of all subsystems that form the augmented system (24).

Specifically, the following set defines the “local input” of the augmented subsystem that is used in the target calculation:

$$\mathbb{I}_i = \{i\} \cup S_i^{OUT} \quad (31)$$

whereas the following set defines its inlet star :

$$\begin{aligned} \bar{S}_i^{IN} &= S_i^{IN} \setminus S_i^{OUT} \\ &= \left[S_i^{IN} \cup S_i^{OUT} \cup \left(\bigcup_{j \in S_i^{OUT}} S_j^{IN} \right) \setminus \{i\} \right] \setminus S_i^{OUT} \\ &= S_i^{IN} \cup \left(\bigcup_{j \in S_i^{OUT}} S_j^{IN} \right) \setminus \mathbb{I}_i \end{aligned} \quad (32)$$

Having defined the following target variables:

$$\bar{x}_{si} = [x_{sj}]_{j \in \mathbb{I}_i}, \quad \bar{y}_{si} = [y_{sj}]_{j \in \mathbb{I}_i}, \quad \bar{u}_{si} = [u_{sj}]_{j \in \mathbb{I}_i} \quad (33)$$

and matrices:

$$\hat{B}_i = \left[\text{hor} \{B_{jk}\}_{k \in \mathbb{I}_i} \right]_{j \in \mathbb{I}_i}, \quad \bar{B}_{ik} = [B_{jk}]_{j \in \mathbb{I}_i} \quad \forall k \in \bar{S}_i^{IN} \quad (34)$$

we can write the equilibrium relation of (24) as:

$$\begin{bmatrix} \bar{A}_i - I & \bar{B}_i & 0 \\ \bar{C}_i & 0 & -I \end{bmatrix} \begin{bmatrix} \bar{x}_{si} \\ \bar{u}_{si} \\ \bar{y}_{si} \end{bmatrix} = \begin{bmatrix} -\sum_{k \in \mathbb{S}_i^{IN}} \bar{B}_{ik} \cdot u_{sk} \\ 0 \end{bmatrix} \quad (35)$$

Notice that the right hand side in (35) contains the input targets for the subsystems belonging to \mathbb{S}_i^{IN} , which are assumed known from previous cooperative iterations. Let the target offset cost of the augmented system be:

$$V_{ss_i}(\bar{y}_{si}, \bar{y}_{ti}) = \|\bar{y}_{si} - \bar{y}_{ti}\|_{\bar{T}_i}^2 \quad (36)$$

and consider the following cost to be minimized by the subsystem i :

$$\begin{aligned} \bar{V}_{ti}(\cdot) = & \sum_{k=0}^{N-1} \bar{\ell}_i(\bar{x}_i(k) - \bar{x}_{si}, u_i(k) - u_{si}) \\ & + \bar{V}_{fi}(\bar{x}_i(k) - \bar{x}_{si}) + V_{ss_i}(\bar{y}_{si}, \bar{y}_{ti}) \quad \text{s.t.} \end{aligned} \quad (37)$$

$$\bar{x}_i(0) = \bar{x}_i$$

$$\bar{x}_i(k+1) = \bar{A}_i \bar{x}_i(k) + \bar{B}_i u_i(k) + \bar{B}_i^{IN} \bar{u}_i(k)$$

Then, the FHOCP in (29) is replaced by

$$\mathbb{P}_i \left(\bar{x}_i, \{ \mathbf{u}_j \}_{j \in \mathbb{S}_i^{IN}}, \{ u_{sj} \}_{j \in \mathbb{S}_i^{IN}} \right) :$$

$$\min_{\mathbf{u}_i, \bar{x}_{si}, \bar{u}_{si}, \bar{y}_{si}} \bar{V}_{ti}(\mathbf{u}_i, \bar{x}_{si}, \bar{u}_{si}, \bar{y}_{si}) \quad (38a)$$

s.t. (35) and

$$\mathbf{u}_i \in \mathcal{U}_i \left(\bar{x}_i, \{ \mathbf{u}_j \}_{j \in \mathbb{S}_i^{IN}} \right) \quad (38b)$$

$$(\bar{x}_i(N), \bar{y}_{si}) \in \bar{\Omega}_i \quad (38c)$$

Once problem (38) is solved to obtain the optimal input sequence \mathbf{u}_i^0 and the optimal augmented input target \bar{u}_{si}^0 , for each subsystem i we can define the overall input target $u_{s,i}^0$, calculated by subsystem i by appending the input targets of all other inputs, i.e. those belonging to $\mathbf{u}_s^{[p-1]}$. Then, the cooperative update of the centralized target is defined as:

$$u_s^{[p]} = \sum_{i=1}^M w_i u_{s,i}^0 \quad (39)$$

A similar update occurs for state and output target $(x_s^{[p]}, y_s^{[p]})$. Likewise, the cooperative update of the local input trajectories is still defined in (16). Furthermore, for the augmented target problem a new tolerance variable ε_{ss} is defined to check convergence of the overall input target $\|u_s^{[p]} - u_s^{[p-1]}\| \leq \varepsilon_{ss}(1 + \|u_s^{[p]}\|)$.

Finally, we can also consider a two step variant algorithm in which the augmented target is optimized instead of the centralized one.

Theorem 6. For any initial condition $x(0)$ and any given set-point y_t , the closed-loop system (1) under the receding horizon control law obtained by Algorithm 1, in which problems \mathbb{P}_i in (38) are solved by each subsystem in place of \mathbb{P}_i in (29), is asymptotically stable and converges to the equilibrium given in (30).

PROOF. Let us define

$$\mathbf{z}^{[p]} = \left(\mathbf{u}_1^{[p]}, \dots, \mathbf{u}_M^{[p]}, x_s^{[p]}, u_s^{[p]}, y_s^{[p]} \right) \quad (40)$$

as the solution of p -th cooperative iteration. If for each i , we solve (38) with $\mathbf{u}_j^{[p]}$, $j \in \mathbb{S}_i$, and $(x_s^{[p]}, u_s^{[p]}, y_s^{[p]})$ fixed, we obtain a solution:

$$\mathbf{z}_i^* = \left(\mathbf{u}_1^{[p]}, \dots, \mathbf{u}_i^*, \dots, \mathbf{u}_M^{[p]}, x_s^{[p]}, u_s^{[p]}, y_s^{[p]} \right) \quad (41)$$

By definition, for each i , it is feasible and it provides a lower cost than $\mathbf{z}^{[p]}$. From convexity of the optimal cost function and the fact that \mathbf{z}_i^0 is obtained from the optimal solution of (38) with $\mathbf{u}_j^{[p]}$ fixed but with $(\bar{x}_{si}, \bar{u}_{si}, \bar{y}_{si})$ as decision variables, we can write:

$$\begin{aligned} V_N(\mathbf{z}^{[p+1]}) &= V_N \left(\sum_{i=1}^M w_i \mathbf{z}_i^0 \right) \\ &\leq \sum_{i=1}^M w_i V_N(\mathbf{z}_i^0) \\ &\leq \sum_{i=1}^M w_i V_N(\mathbf{z}_i^*) \\ &\leq \sum_{i=1}^M w_i V_N(\mathbf{z}^{[p]}) = V_N(\mathbf{z}^{[p]}) \end{aligned} \quad (42)$$

This shows that:

$$V_N(\mathbf{z}^{[p+1]}) \leq V_N(\mathbf{z}^{[p]}), \quad \forall p \in \mathbb{N} \quad (43)$$

The rest of the proof follows closely that of Th. 1 in [15].

3.5. Complexity analysis

It is important to remark that in problem (29), 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 (8) or (29), are more effectively solved for large scale systems using Interior Point algorithms with both state and input sequences as decision variables [17]. 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 [15]. However, in the steady-state problem there is still the centralized target triple

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

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

(x_s, u_s, y_s) . As discussed in §3.4, thanks to the augmented target model formulation, also the complexity of computation for steady steady targets can be reduced.

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 standard method discussed in §2.4, DMPC1-CT is the proposed single step method discussed in §3.2, DMPC2-CT is the proposed two step method discussed in §3.3. DMPC1-AT and DMPC2-AT are the proposed one or two step methods exposed in §3.4.

4. Distributed offset-free tracking

In the previous sections, we assumed that the current state of each subsystem x_i was measurable and that the true controlled process followed the nominal DLTI dynamics given in (1). In this section, both requirements are relaxed assuming that each subsystem i – measures only the local output y_i at each sampling time, and that the true controlled process may not follow (1) precisely. Nonetheless, we still aim to achieve setpoint tracking, i.e.

$$\lim_{k \rightarrow \infty} y_i(k) = y_{i}, \quad \forall i \in \{1, \dots, M\} \quad (44)$$

For the scope of this section, Assumption 1 is replaced by the following one.

Assumption 7. For each subsystem i : the output y_i is measurable at each decision time, the pair (A_i, B_i) is controllable, the pair (C_i, A_i) is observable, and the following condition holds true:

$$\text{rank} \begin{bmatrix} A_i - I & B_i \\ C_i & 0 \end{bmatrix} = n_i + p_i \quad (45)$$

4.1. Distributed offset-free estimation

Following the well established theory of (centralized) offset-free MPC [18, 19, 20, 21], for each subsystem i we use an observer to estimate the augmented state of the following system:

$$\begin{aligned} x_i^+ &= A_i x_i + B_i u_i + \sum_{j \in \mathcal{N}_i} B_{ij} u_j + B_{di} d_i \\ d_i^+ &= d_i \\ y_i &= C_i x_i + C_{di} d_i \end{aligned} \quad (46)$$

in which $d_i \in \mathbb{R}^{p_i}$ is the so-called ‘‘integral’’ disturbance, and the matrices $B_{di} \in \mathbb{R}^{n_i \times p_i}$ and $C_{di} \in \mathbb{R}^{p_i \times p_i}$ define the disturbance

model, i.e the way in which the disturbance affects the state-transition map and the output map. In addition to Assumption 7, the following further assumption is made.

Assumption 8. For each subsystem i , the disturbance model matrices (B_{di}, C_{di}) are such that the following condition holds true:

$$\text{rank} \begin{bmatrix} A_i - I & B_{di} \\ C_i & C_{di} \end{bmatrix} = n_i + p_i \quad (47)$$

Notice that there always exist pairs (B_{di}, C_{di}) satisfying the above requirement if and only if (C_i, A_i) is observable.

Any (linear) observer can be used, and for the sake of simplicity we here focus on the use of a static gain matrix $K_i \in \mathbb{R}^{(n_i + p_i) \times p_i}$, partitioned as:

$$K_i = \begin{bmatrix} K_{xi} \\ K_{di} \end{bmatrix} \quad (48)$$

which is used to update the prediction of the local state and disturbance, denoted by $(\hat{x}_i(k|k-1), \hat{d}_i(k|k-1))$, as follows:

$$\begin{aligned} \begin{bmatrix} \hat{x}_i(k|k) \\ \hat{d}_i(k|k) \end{bmatrix} &= \begin{bmatrix} \hat{x}_i(k|k-1) \\ \hat{d}_i(k|k-1) \end{bmatrix} + \\ &\begin{bmatrix} K_{xi} \\ K_{di} \end{bmatrix} \left(y_i(k) - [C_i \ C_{di}] \begin{bmatrix} \hat{x}_i(k|k-1) \\ \hat{d}_i(k|k-1) \end{bmatrix} \right) \end{aligned} \quad (49)$$

The filter gain matrix must satisfy the stability condition that the following matrix:

$$\begin{bmatrix} A_i & B_{di} \\ 0 & I \end{bmatrix} - \begin{bmatrix} A_i & B_{di} \\ 0 & I \end{bmatrix} \begin{bmatrix} K_{xi} \\ K_{di} \end{bmatrix} [C_i \ C_{di}] \quad (50)$$

is Hurwitz. Under Assumptions 7 and 8, the existence of such $\begin{bmatrix} K_{xi} \\ K_{di} \end{bmatrix}$ is guaranteed, e.g. by designing a steady-state Kalman filter for (46) or via pole-placement.

4.2. Optimal control problem for offset-free tracking

Given the current state and disturbance estimate of each subsystem $(\hat{x}_i(k|k), \hat{d}_i(k|k))$, we build the augmented system used for the local MPC problem according to the procedure described in §3.1, i.e. defining the local augmented system state, input and output as in (22). Consequently the augmented disturbance vector is defined as:

$$\bar{d}_i = \begin{bmatrix} \hat{d}_i(k|k) \\ [\hat{d}_j(k|k)]_{j \in \mathcal{S}_i^{OUT}} \end{bmatrix}, \quad (51)$$

and the augmented disturbance model matrices are defined as:

$$\begin{aligned}\bar{B}_{di} &= \text{diag} \left\{ B_{di}, \{B_{dj}\}_{j \in \mathcal{S}_i^{\text{OUT}}}\right\}, \\ \bar{C}_{di} &= \text{diag} \left\{ C_{di}, \{C_{dj}\}_{j \in \mathcal{S}_i^{\text{OUT}}}\right\}\end{aligned}\quad (52)$$

With these definitions, the prediction model used by each local MPC is given by:

$$\begin{aligned}\bar{x}_i^+ &= \bar{A}_i \bar{x}_i + \bar{B}_i u_i + \bar{B}_i^{\text{IN}} \bar{u}_i + \bar{B}_{di} \bar{d}_i \\ \bar{y}_i &= \bar{C}_i \bar{x}_i + \bar{C}_{di} \bar{d}_i\end{aligned}\quad (53)$$

In order to define the centralized target model, we stack together the disturbance model matrices (B_{di}, C_{di}) and the disturbance estimate $\hat{d}_i(k|k)$ each subsystem (46):

$$\begin{aligned}B_d &= \text{diag} \{B_{di}\}_{i=1}^M, \quad C_d = \text{diag} \{C_{di}\}_{i=1}^M, \\ \hat{d}(k|k) &= [\hat{d}_i(k|k)]_{i=1}^M\end{aligned}\quad (54)$$

Thus, we can write the steady-state centralized offset-free model as:

$$\begin{bmatrix} A-I & B & 0 \\ C & 0 & -I \end{bmatrix} \begin{bmatrix} x_s \\ u_s \\ y_s \end{bmatrix} = - \begin{bmatrix} B_d \\ C_d \end{bmatrix} \hat{d}(k|k)\quad (55)$$

Then, the FHOCPC that is solved at each sampling time reads:

$$\mathbb{P}_i \left(\bar{x}_i, \{\mathbf{u}_j\}_{j \in \mathcal{S}_i^{\text{IN}}} \right) : \min_{\mathbf{u}_i, x_s, u_s, y_s} V_{ti}(\mathbf{u}_i, x_s, u_s, y_s)\quad (56a)$$

s.t. (55) and

$$\mathbf{u}_i \in \bar{\mathcal{U}}_i \left(\bar{x}_i, \{\mathbf{u}_j\}_{j \in \mathcal{S}_i^{\text{IN}}} \right)\quad (56b)$$

$$(\bar{x}_i(N), \bar{y}_{si}) \in \bar{\Omega}_i\quad (56c)$$

in which, recalling the prediction model (53), we have:

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

Once we solve problem (56) and we obtain an optimal input sequence \mathbf{u}_i^0 , the usual cooperative iterations are performed.

Obviously, we can consider a two step variant, as well as the augmented target model approach instead of the centralized one in (55).

Theorem 9. *Assume that the closed-loop system reaches an equilibrium with input u_∞ and output y_∞ , and that state and input constraints are inactive at this equilibrium. Let \hat{d}_∞ be the corresponding steady value reached by the disturbance estimate. It follows that the output of the controlled system y_∞ is equal to the best reachable target $y_{s,\infty}$ corresponding to the solution of:*

$$\min_{(x_s, u_s, y_s)} V_{ss}(y_s, y_t) \quad \text{s.t.}\quad (57a)$$

$$x_s = Ax_s + Bu_s + B_d \hat{d}_\infty\quad (57b)$$

$$y_s = Cx_s + C_d \hat{d}_\infty\quad (57c)$$

PROOF. Consider the centralized offset-free model

$$\begin{aligned}x^+ &= Ax + Bu + B_d d \\ d^+ &= d \\ y &= Cx + C_d d\end{aligned}\quad (58)$$

which is obtained by combining together the local offset-free models (46) for all i . The overall disturbance predictor is:

$$\hat{d}_{k+1|k} = \hat{d}_{k|k-1} + K_d (y_k - C \hat{x}_{k|k-1} - C_d \hat{d}_{k|k-1})\quad (59)$$

in which $K_d = \text{diag} \{K_{di}\}_{i=1}^M$. From (59) and the results in [19], each K_{di} is invertible, and therefore K_d is also invertible. Since the system has reached an equilibrium, from (59) it follows that:

$$\lim_{k \rightarrow \infty} (y_k - C \hat{x}_{k|k-1} - C_d \hat{d}_{k|k-1}) = 0\quad (60)$$

or equivalently:

$$y_\infty = C \hat{x}_\infty + C_d \hat{d}_\infty\quad (61)$$

in which:

$$\begin{aligned}\hat{d}_\infty &= \lim_{k \rightarrow \infty} \hat{d}(k|k-1) = \lim_{k \rightarrow \infty} \hat{d}(k|k), \\ \hat{x}_\infty &= \lim_{k \rightarrow \infty} \hat{x}(k|k-1) = \lim_{k \rightarrow \infty} \hat{x}(k|k)\end{aligned}\quad (62)$$

Given that the disturbance reached the constant value \hat{d}_∞ , for each subsystem the centralized equilibrium target triple (x_s, u_s, y_s) is also constant. We now show that $\hat{x}_\infty = x_{s,\infty}$ solution to (57). Assume, by contradiction that $\hat{x}_\infty \neq x_{s,\infty}$, and recall that state and input constraints are inactive at the reached equilibrium. It would follow that $u_{i,\infty} = u_i^0(0) \neq u_{si,\infty}$. Each subsystem solves (56) using (virtually) the centralized dynamics, and hence we would have:

$$\hat{x}_\infty = A \hat{x}_\infty + B u_\infty + B_d \hat{d}_\infty\quad (63)$$

along the prediction horizon. However, the stage cost $\ell(\cdot)$ is strictly convex in its arguments and constraints are inactive. This means that cost function returns, due to (63), a value different from the minimum, which is a contradiction. Hence, the solution computed by each subsystem is

$$\begin{aligned}u_\infty &= u_{s,\infty} \\ \hat{x}_\infty &= x_{s,\infty}\end{aligned}\quad (64)$$

From (64) the following holds:

$$y_{s,\infty} = C x_{s,\infty} + C_d \hat{d}_\infty = y_\infty\quad (65)$$

5. Application examples

5.1. Multi-stage evaporator model and subsystems

As an illustrative example, we consider a ‘‘forward feed’’, triple effect, evaporator process. A simplified process flow diagram is shown in Figure 3. The mass and energy balance

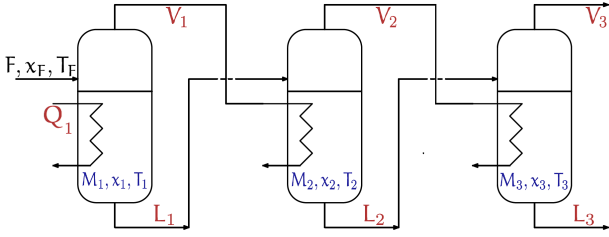


Figure 3: Simplified process flow diagram (PFD) of the "forward feed", triple effect, evaporator process.

equations are different in the first or in subsequent ($i = 2, 3$) evaporators. For the first one we have:

$$\begin{cases} \frac{dM_1}{dt} = F - L_1 - V_1 \\ c_p M_1 \frac{dT_1}{dt} = F c_p (T_F - T_1) - V_1 \lambda_1 + Q_1 \\ M_1 \frac{d\chi_1}{dt} = F \chi_F + (V_1 - F) \chi_1 \end{cases} \quad (66)$$

in which the outputs are the liquid mass M_1 , the evaporator temperature T_1 , the solute mass fraction χ_1 , whereas the inputs are the liquid outlet rate L_1 , the external duty Q_1 , the vapor outlet rate V_1 . The other parameters are the feed rate F and temperature T_F , the liquid specific heat c_p , the heat of evaporation λ_1 . The balance equations for subsequent evaporators ($i = 2, 3$), instead, read:

$$\begin{cases} \frac{dM_i}{dt} = L_{i-1} - L_i - V_i \\ c_p M_i \frac{dT_i}{dt} = L_{i-1} c_p (T_{i-1} - T_i) - V_i \lambda_i + V_{i-1} \lambda_{i-1} \\ M_i \frac{d\chi_i}{dt} = L_{i-1} \chi_{i-1} + (V_i - L_{i-1}) \chi_i \end{cases} \quad (67)$$

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 . The heat of evaporation is λ_i . Model parameters are taken from [22, p.632-633].

To obtain a linear model, the nonlinear system (66)-(67) is tested for 72 hours in which each input has 2% of maximum amplitude variation. Generalized Binary Noise (GBN) input signals are produced [23] with a sampling time of 1 min and 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 [24]. 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.

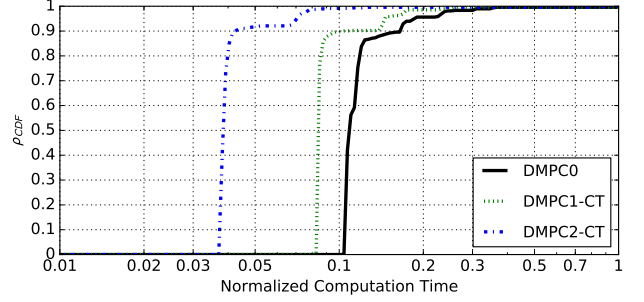


Figure 4: Cumulative distribution function of normalized computation time of DMPC1-CT (one step, centralized target model) and DMPC2-CT (two steps, centralized target model).

5.2. Comparison of different controllers in nominal case

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 targets on the three solute mass fraction:

- $k = 5, \dots, 239: \chi_t = [13.4, 23.2, 52] \%$;
- $k = 240, \dots, 480: \chi_t = [13.2, 22.2, 47] \%$.

Three controllers are compared: the first one is DMPC0, associated with the single step algorithm by [15] as discussed in (§2.4); the second one is the proposed DMPC1-CT (§3.2); the third is the proposed two steps algorithm DMPC2-CT (§3.3). All controllers are tuned with the same parameters:

- $N = 100, \|u\|_\infty \leq 1$;
- $Q_i = C_i^T \text{diag}([0.1, 0.1, 1]) C_i$, for $i = 1, 2, 3$;
- $R_1 = 0.01 I_3, R_i = 0.01 I_2$ for $i = 2, 3$;
- $T_1 = 10^6 \cdot I_3, T_i = \text{diag}([10^6, 10^{-3}, 10^6])$ 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 till the maximum number of cooperative iteration ($q_{\max} = 100$) are reached or convergence error is less than $\varepsilon = 0.01$;
- $\varepsilon_{ss} = 0.0001$.

Simulations are performed on a code written in Python 2.7, with the use of symbolic framework offered by CasADi 3.1. All optimization modules implement IPOPT.

Figure 4 shows the cumulative distribution function (CDF) of computation times, normalized by the maximum computation time occurred among all controllers. 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-CT completed about 90% of its executions in less than 5% of the maximum computation time, while DMPC1-CT completed the same percentage of executions in about 10% of the maximum computation time. On the other hand, DMPC0 completed no executions in 10% of the maximum computation time; at least

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	$\frac{-0.01013z+0.01865}{z^2-5.241z+6.864}$	-	$\frac{0.004064z-0.005355}{z^2-5.241z+6.864}$	$\frac{-0.2224z+0.6029}{z^2-5.241z+6.864}$	$\frac{0.01244z-0.02893}{z^2-5.241z+6.864}$	-	$\frac{0.464z-1.249}{z^2-5.241z+6.864}$

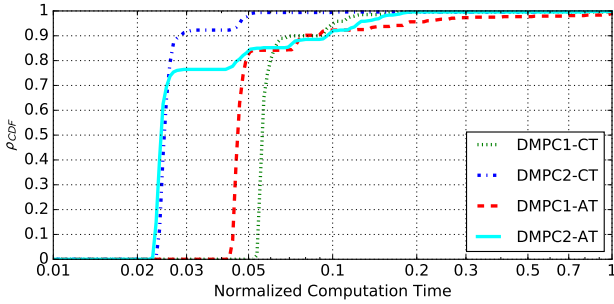


Figure 5: Cumulative distribution function of normalized computation time of DMPC1-CT (one step, centralized target model), DMPC2-CT (two steps, centralized target model), DMPC1-AT (one step, augmented target model), and DMPC2-AT (two steps, augmented target model).

it needs about 20% of the maximum time to complete 90% of its executions. From this plot we can appreciate how DMPC1-CT and DMPC2-CT are parsimonious with respect to DMPC0.

Figure 5 shows CDF comparison between centralized tracking as computed in DMPC1-CT and DMPC2-CT with respect to DMPC1-AT and DMPC2-AT, which implement an augmented target model as described in §3.4, respectively in single step and two steps. We observe that DMPC1-AT is able to complete about 85% of executions faster than DMPC1-CT, while in the remaining 15% of executions DMPC1-AT is slower than DMPC1-CT. A similar outcome can be noticed in the comparison between DMPC2-AT and DMPC2-CT. This behavior can be explained by noticing that in those cases, more cooperative iterations may be required to meet the given tolerance when an augmented target model is used in a tracking MPC algorithm. Thus, despite the fact that at each iteration a smaller problem is solved, the overall computation time may increase. Thus, a fully distributed tracking MPC algorithm appears more effective if one poses a limit on the maximum number of cooperative iterations, hence with a possible loss of optimality.

Figures 6 and 7 show the closed-loop evolution of inputs (with constraints) and outputs with relative targets. Results are similar for all controllers, and DMPC0 and DMPC1 are (as

expected) identical in closed-loop performance. By using the augmented state rather than the centralized one in the prediction model, we have not taken out useful information from the overall dynamics.

5.3. Offset-free performance comparison in the presence of disturbances

In this case, we keep the desired setpoint constant, but the system is affected by two input step disturbances, one entering at time 5 min, and the other one at time 4 hr. Two controllers are here compared: DMPC1-AT (single-step, augmented dynamic model, augmented target model), DMPC1-AT-OF (single-step, offset-free augmented dynamic model, offset-free augmented target model). All tuning parameters are the same as those used in the nominal case. We use a mixed input/output disturbance model as discussed in [21]. So we define B_{di} , as a discrete time steady-state Kalman filter gain computed for the given discrete time invariant system in state space form (numerically solved by a Riccati equation) with appropriate process and measurements noise covariance matrices $Q_x = 0.001I$ and $R_y = 0.01I$. Then we define C_{di} as

$$C_{di} = I - C_i B_{di}$$

Disturbance observer is then computed as Kalman filter for the given linear system in state space form (Luenberger observer).

Figures 8 and 9 show the closed-loop evolution of inputs (with constraints) and outputs with relative desired setpoint. As expected, the offset-free distributed MPC is able to reject the unknown disturbances and achieve asymptotic setpoint tracking.

6. Conclusions

We presented in this paper several 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 dynamic prediction and steady-state models

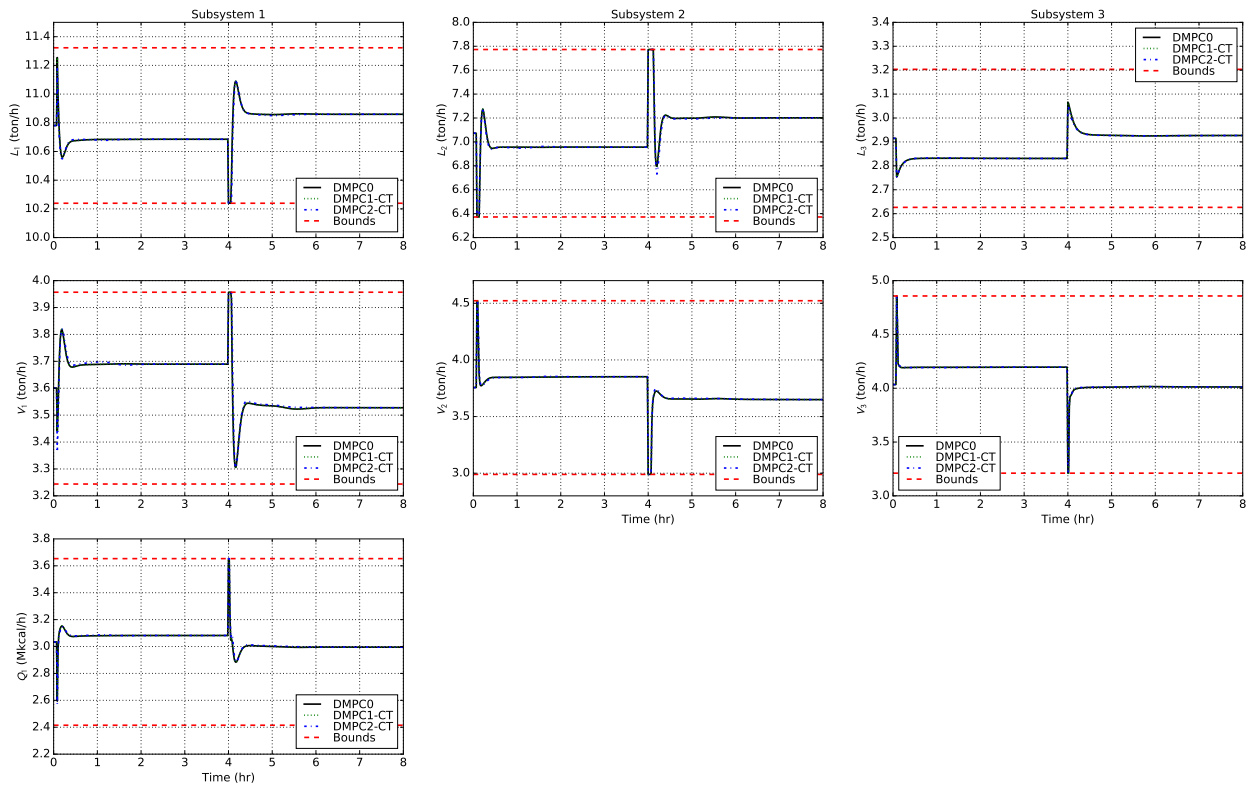


Figure 6: Closed-loop evolution of inputs in the nominal case. Red dashed lines are used for input bounds.

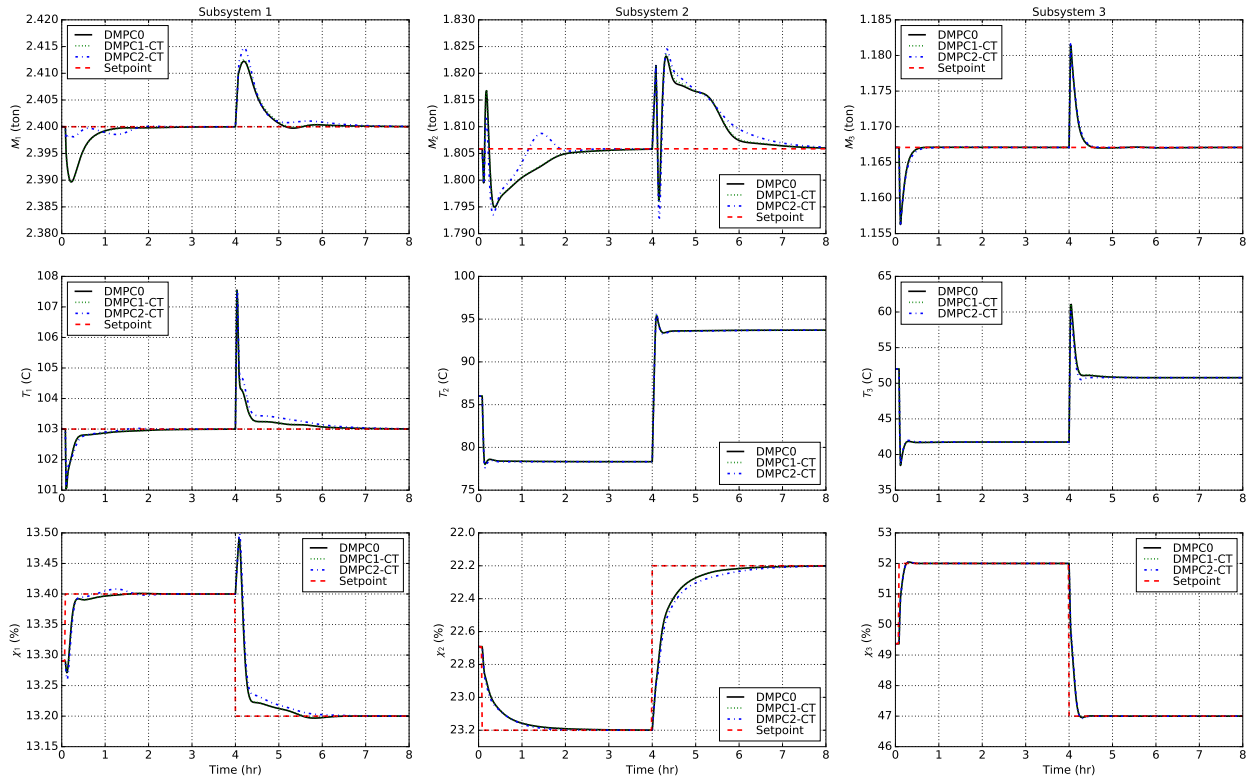


Figure 7: Closed-loop evolution outputs in the nominal case. Red dashed lines are used for output setpoints.

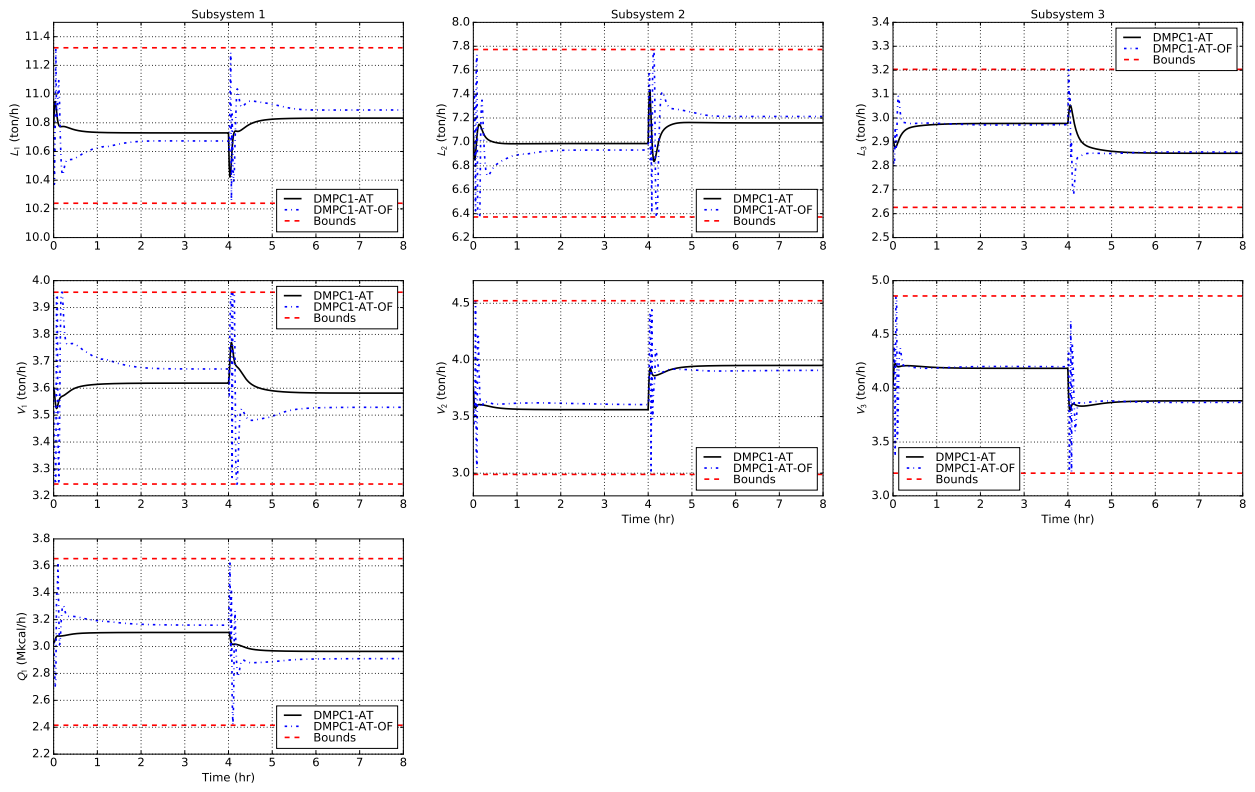


Figure 8: Closed-loop evolution of inputs in the presence of disturbances. Red dashed lines are used for input bounds.

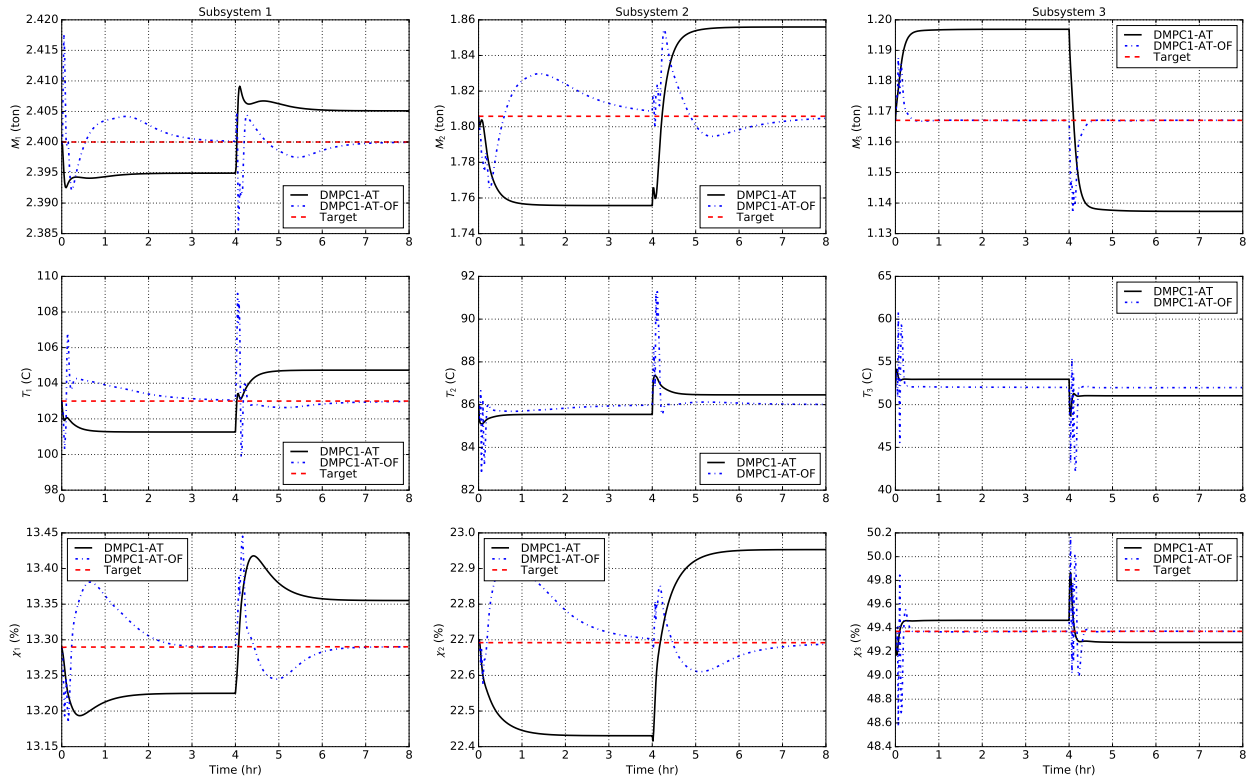


Figure 9: Closed-loop evolution outputs in the presence of disturbances. Red dashed lines are used for output setpoints.

used by each subsystem, by exploiting concepts of graph theory, while retaining the global optimality of the cooperative algorithm. This approach reduces the computational and communication requirements of the proposed algorithms with respect to the currently available ones. Two basic variants were considered, one with target calculation embedded in the optimal control problem and one with separate target calculation. Then, we showed that the target model can be centralized or in augmented version. Furthermore, we presented the extension of the proposed algorithms to account for incomplete state-measurement and presence of plant/model mismatch, designing for each subsystem a local offset-free observer.

A multiple effect evaporator process composed by three subsystems (overall with 12 states, 7 inputs, and 9 outputs) has been presented, and the proposed algorithms were shown to be more computationally efficient than available methods, as well as they are able to eliminate steady-state offset in the presence of unknown persistent disturbances.

The proposed algorithms will be further analyzed to provide the main theoretical guarantees, especially their inherent robustness [4], and explore if/how it is possible to derive nonlinear distributed MPC algorithms for tracking.

References

- [1] M. Razzanelli, G. Pannocchia, Parsimonious cooperative distributed MPC for tracking piece-wise constant setpoints, in: *IFAC-PapersOnLine*, Vol. 49, 2016, pp. 520–525.
- [2] M. J. Maciejowski, *Predictive Control with Constraints*, Prentice Hall, 2000.
- [3] B. J. Rawlings, Q. D. Mayne, *Model Predictive Control: Theory and Design*, Nob Hill Publishing, 2009.
- [4] G. Pannocchia, J. B. Rawlings, S. J. Wright, Conditions under which sub-optimal nonlinear MPC is inherently robust, *Syst. Contr. Lett.* 60 (2011) 747–755.
- [5] D. Limon, I. Alvarado, T. Alamo, E. Camacho, MPC for tracking piece-wise constant references for constrained linear systems, *Automatica* 44 (2008) 2382–2387.
- [6] P. Trodden, Cooperative distributed MPC of linear systems with coupled constraints, *Automatica* 49 (2013) 479–487.
- [7] P. Trodden, A. Richards, Distributed model predictive control of linear systems with persistent disturbances, *International Journal of Control* 83 (2010) 1653–1663.
- [8] A. Bemporad, D. Barcelli, Decentralized model predictive control, in: A. Bemporad, M. Heemels, M. Johansson (Eds.), *Networked Control Systems*, Springer Lecture Notes in Control and Information Sciences, Springer, 2010.
- [9] S. Rivero, M. Farina, G. Ferrari-Trecate, Plug-and-play decentralized model predictive control for linear systems, *IEEE Trans. Auto. Contr.* (2013) 2608–2614.
- [10] R. Scattolini, Architectures for distributed and hierarchical model predictive control - a review, *Journal of Process Control* 19 (2009) 723–731.
- [11] M. Kögel, R. Findeisen, Set-point tracking using distributed MPC, *IFAC Proceedings Volumes* 46 (32) (2013) 57–62.
- [12] M. Farina, R. Scattolini, Distributed predictive control: A non-cooperative algorithm with neighbor-to-neighbor communication for linear systems, *Automatica* 48 (2012) 1088–1096.
- [13] G. Betti, M. Farina, R. Scattolini, Distributed MPC: A noncooperative approach based on robustness concepts., in: J. M. Maestre, R. R. Negenborn (Eds.), *Distributed Model Predictive Control Made Easy*, Springer, Upper Saddle River, New Jersey, 2014, Ch. 26.
- [14] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, G. Pannocchia, Cooperative distributed model predictive control, *Syst. Contr. Lett.* 59 (2010) 460–469.
- [15] A. Ferramosca, D. Limon, I. Alvarado, E. F. Camacho, Cooperative distributed MPC for tracking, *Automatica* 49 (2013) 906–914.
- [16] G. Pannocchia, Distributed model predictive control, in: B. T. Samad (Ed.), *Encyclopedia of Systems and Control*, Springer, 2014, Ch. 5, pp. 302–308.
- [17] C. V. Rao, S. J. Wright, J. B. Rawlings, Application of interior-point methods to model predictive control, *J. Optim. Theory Appl.* 99 (1998) 723–757.
- [18] K. R. Muske, T. A. Badgwell, Disturbance modeling for offset-free linear model predictive control, *J. Proc. Contr.* 12 (2002) 617–632.
- [19] G. Pannocchia, J. B. Rawlings, Disturbance models for offset-free model predictive control, *AIChE J.* 49 (2003) 426–437.
- [20] U. Maeder, F. Borrelli, M. Morari, Linear offset-free model predictive control, *Automatica* 45 (10) (2009) 2214–2222.
- [21] G. Pannocchia, Offset-free tracking MPC: A tutorial review and comparison of different formulations, in: *Proceedings of European Control Conference, ECC 2015*, 2015, pp. 527–532.
- [22] J. M. Coulson, J. F. Richardson, *Chemical Engineering*, Vol. 2, Pergamon Press, 1993.
- [23] Y. Zhu, *Multivariable System Identification for Process Control*, Springer, 2001.
- [24] L. Ljung, *System Identification: Theory for the User*, 2nd Edition, Prentice Hall Inc., Upper Saddle River, New Jersey, 1999.