A review on distributed predictive control: basic ideas, extensions, and applications

Marcello Farina *, Giulio Betti, Luca Giulioni, Riccardo Scattolini

Dipartimento di Elettronica, Informazione e Bioingengeria, Politecnico di Milano, Piazza Leonardo da Vinci, 32, I-20133 Milan, Italy

Abstract

This paper presents an overview on the Distributed Predictive Control (DPC) algorithm first proposed in [15]. First we present the basic ideas behind DPC and the main assumptions and then we focus on its main properties, extensions, and application examples developed and analyzed in subsequent works, e.g., [13,3,11,10,5,6], and consistently presented in [2].

1 Introduction

Due to the growing complexity of process plants and to the increasing number of networks of systems, in the last decades researchers have been putting huge efforts in the field of decentralized and distributed control [27,18]. Distributed solutions seem to be very promising with respect to decentralized schemes, because they allow one to take advantage of the possibility to transmit information between the local controllers, see, e.g., [17], and do not require the computational and communication loads of centralized solutions. However, distributed techniques are characterized by an intrinsically higher degree of complexity in the design phase with respect to centralized controllers. This could represent a great obstacle to their diffusion in the industrial world, and motivates the development of many innovative distributed Model Predictive Control (MPC) algorithms for large-scale systems, see the survey papers [25,7] and the book [22], where the most recent and popular algorithms have been collected and described.
According to the classification proposed in [25], a new non-iterative, non-cooperative approach based on neighbor-to-neighbor communication, called Distributed Predictive Control (DPC), has been proposed in [15], where its convergence and stability properties have also been extensively analyzed. The highlights of DPC are the following.

- It is not necessary for each subsystem to know the dynamical models governing the trajectories of the other subsystems (not even the ones of its neighbors), leading to a non-cooperative approach.
- The transmission of information is limited (i.e., DPC is non-iterative [25] and requires a neighbor-to-neighbor communication network), in that each subsystem needs to know the reference trajectories only of its neighbors.
- Its rationale is similar to the MPC algorithms often employed in industry: reference trajectories tailored on the dynamics of the system under control are used.
- Convergence and stability properties are guaranteed under mild assumptions.

For a practical application of DPC, a number of issues concerning its realization and tuning have been tackled in subsequent papers [5,6], where its performances have also been assessed in realistic simulation scenarios. A number of papers have also focused on the extension of DPC to output feedback control [14] and to the problem of tracking constant reference outputs, e.g., [3,11,10]. The realization issues, the extensions and the main applications of DPC are consistently and thoroughly discussed and presented in the PhD Thesis [2].
In this paper the state feedback Distributed Predictive Control (DPC) algorithm originally proposed in [15] is sketched and discussed, as well as its properties and extensions. Finally, some significant realistic simulations and a real experimental test are illustrated, to highlight the applicability of the proposed algorithm.

The paper is organized as follows: in Sections 2 and 3 the main control problem is stated and the DPC algorithm is summarized, respectively. In Section 4 the main properties and extensions of DPC are discussed, while in Section 5 we present a number of application examples. Finally, some conclusions are drawn in Section 6.

Notation. A matrix is Schur stable if all its eigenvalues lie in the interior of the unit circle. The short-hand $\mathbf{v} = (v_1, \dots, v_s)$ denotes a column vector with s (not necessarily

* Corresponding author: Tel.: +39-02.23993599; fax: 39-02.23993412.
E-mail address: marcello.farina@polimi.it

scalar) components $v_1, \ldots, v_s$. The symbol $\oplus$ denotes the Minkowski sum, namely $C = A \oplus B$ if and only if $C = \{c : c = a + b, \text{ for all } a \in A, b \in B\}$, while $\bigoplus_{i=1}^{M} A_i = A_1 \oplus \cdots \oplus A_M$. The Pontryagin difference is defined using the symbol $\ominus$, i.e. $C = A \ominus B$ if and only if $C = \{c : c + b \in A, \text{ for all } b \in B\}$. For a discrete-time signal s_t and $a, b \in \mathbb{N}$, $a \leq b$, we denote $(s_a, s_{a+1}, \ldots, s_b)$ with $s_{[a:b]}$.

2 Statement of the problem and main assumptions

In this section, the Distributed Predictive Control (DPC) algorithm first presented in [15] and further developed in, e.g., [5,6] is briefly described. Let us assume that the system is constituted by M linear, discrete-time, non-overlapping subsystems, dynamically coupled through states and inputs, and subject to state and control constraints. For each subsystem $\mathscr{S}_i$, the dynamics is given by

$$\mathbf{x}_{k+1}^{[i]} = \mathbf{A}_{ii}\mathbf{x}_k^{[i]} + \mathbf{B}_{ii}\mathbf{u}_k^{[i]} + \sum_{j=1, j\neq i}^{M} \{\mathbf{A}_{ij}\mathbf{x}_k^j + \mathbf{B}_{ij}\mathbf{u}_k^j\} + \mathbf{d}_k^{[i]} \quad (1)$$

where $\mathbf{x}^{[i]} \in \mathscr{X}_i \subseteq \mathscr{R}^{n_i}$ and $\mathbf{u}^{[i]} \in \mathscr{U}_i \subseteq \mathscr{R}^{m_i}$ are the state and input vectors of the i-th subsystem $\mathscr{S}_i$ ($i = 1, ..., M$), $\mathbf{d}^{[i]} \in \mathscr{D}_i \subset \mathscr{R}^{n_i}$ is an unknown bounded disturbance and the sets $\mathscr{X}_i$, $\mathscr{U}_i$ and $\mathscr{D}_i$ are convex neighborhoods of the origin. The subsystem $\mathscr{S}_j$ is said to be a *neighbor* of the subsystem $\mathscr{S}_i$ if and only if $\mathbf{A}_{ij} \neq 0$ and/or $\mathbf{B}_{ij} \neq 0$, i.e., if and only if the states x_j and/or inputs u_j of $\mathscr{S}_j$ influence the dynamics of $\mathscr{S}_i$. The symbol $\mathscr{N}_i$ denotes the set of neighbors of $\mathscr{S}_i$ (which excludes i). Note that also constraints involving the state of more than one subsystem at the same time can be accounted for. However, for simplicity of presentation, they are discarded in the present paper. For details see [15,5].

Letting $\mathbf{x}_k = (\mathbf{x}_k^{[1]}, \ldots, \mathbf{x}_k^{[M]})$, $\mathbf{u}_k = (\mathbf{u}_k^{[1]}, \ldots, \mathbf{u}_k^{[M]})$ and $\mathbf{d}_k = (\mathbf{d}_k^{[1]}, \ldots, \mathbf{d}_k^{[M]})$, the overall collective system can be written as

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{d}_k \quad (2)$$

where the matrices $\mathbf{A}$ and $\mathbf{B}$ have block entries $\mathbf{A}_{ij}$ and $\mathbf{B}_{ij}$ respectively, $\mathbf{x} \in \mathscr{X} = \prod_{i=1}^{M} \mathscr{X}_i \subseteq \mathscr{R}^n$, $n = \sum_{i=1}^{M} n_i$, $\mathbf{u} \in \mathscr{U} = \prod_{i=1}^{M} \mathscr{U}_i \subseteq \mathscr{R}^m$, $m = \sum_{i=1}^{M} m_i$, $\mathbf{d} \in \mathscr{D} = \prod_{i=1}^{M} \mathscr{D}_i \subset \mathscr{R}^n$, and $\mathscr{X}$, $\mathscr{U}$ are convex by convexity of $\mathscr{X}_i$ and $\mathscr{U}_i$, respectively. The following assumption on decentralized stabilizability is needed.

Assumption 1 *There exists a block diagonal matrix* $\mathbf{K} = diag(\mathbf{K}_1, \ldots, \mathbf{K}_M)$, *with* $\mathbf{K}_i \in R^{m_i \times n_i}$, $i = 1, \ldots, M$ *such that: (i)* $\mathbf{A} + \mathbf{B}\mathbf{K}$ *is Schur, (ii)* $\mathbf{F}_{ii} = (\mathbf{A}_{ii} + \mathbf{B}_{ii}\mathbf{K}_i)$ *is Schur,* $i = 1, \ldots, M$.

3 Description of the approach

At any time instant k, each subsystem $\mathscr{S}_i$ transmits to its neighbors its future state and input reference trajectories (to be later specified) defined over the prediction horizon N, and called $\tilde{\mathbf{x}}_{k+\nu}^{[i]}$ and $\tilde{\mathbf{u}}_{k+\nu}^{[i]}$, $\nu = 0, \ldots, N-1$, respectively. These trajectories coincide with the *assumed trajectories* introduced in [9]. By adding suitable constraints to its MPC formulation, $\mathscr{S}_i$ is able to guarantee that, for all $k \geq 0$, its real trajectories lie in specified time invariant neighborhoods of $\tilde{\mathbf{x}}^{[i]}$ and $\tilde{\mathbf{u}}^{[i]}$, i.e., $\mathbf{x}_k^{[i]} \in \tilde{\mathbf{x}}_k^{[i]} \oplus \mathscr{E}_i$ and $\mathbf{u}_k^{[i]} \in \tilde{\mathbf{u}}_k^{[i]} \oplus \mathscr{E}_i^{\mathscr{U}}$, where $0 \in \mathscr{E}_i$ and $0 \in \mathscr{E}_i^{\mathscr{U}}$. In this way, the dynamics (1) of $\mathscr{S}_i$ can be written as

$$\mathbf{x}_{k+1}^{[i]} = \mathbf{A}_{ii}\mathbf{x}_k^{[i]} + \mathbf{B}_{ii}\mathbf{u}_k^{[i]} + \sum_{j\in\mathscr{N}_i} \{\mathbf{A}_{ij}\tilde{\mathbf{x}}_k^j + \mathbf{B}_{ij}\tilde{\mathbf{u}}_k^j\} + \mathbf{w}_k^{[i]} \quad (3)$$

where

$$\mathbf{w}_k^{[i]} = \sum_{j\in\mathscr{N}_i} \{\mathbf{A}_{ij}(\mathbf{x}_k^j - \tilde{\mathbf{x}}_k^j) + \mathbf{B}_{ij}(\mathbf{u}_k^j - \tilde{\mathbf{u}}_k^j)\} + \mathbf{d}_k^{[i]} \in \mathscr{W}_i$$

and $\mathscr{W}_i = \bigoplus_{j\in\mathscr{N}_i} \{\mathbf{A}_{ij}\mathscr{E}_j \oplus \mathbf{B}_{ij}\mathscr{E}_j^{\mathscr{U}}\} \oplus \mathscr{D}_i$.

The main idea behind DPC is that each subsystem solves a robust MPC optimization problem considering that its dynamics is given by (3), where the term $\sum_{j\in\mathscr{N}_i}(\mathbf{A}_{ij}\tilde{\mathbf{x}}_{k+\nu}^j + \mathbf{B}_{ij}\tilde{\mathbf{u}}_{k+\nu}^j)$ can be interpreted as an input known in advance over the prediction horizon $\nu = 0, \ldots, N-1$ to be suitably compensated and $\mathbf{w}_k^{[i]}$ is a bounded disturbance to be rejected. By definition, $\mathbf{w}_k^{[i]}$ represents the uncertainty of the future actions that will be carried out by the dynamic neighbors of subsystem S_i. Therefore the local MPC optimization problem to be solved at each time instant by the controller embedded in subsystem S_i must minimize the cost associated to S_i for any possible uncertainty values, i.e., without having to make any assumption on the strategies adopted by the other subsystems, provided that their future trajectories lie in the specified neighborhood of the reference ones. Such *conservative* but *robust* local strategies adopted by each subsystem can be interpreted, from a dynamic non-cooperative game theoretic perspective, as maxmin strategies, i.e., the strategies that maximize "worst case utility" of S_i (for more details see, e.g., [26]).
To solve local robust MPC problems (denoted i-DPC problems), the algorithm proposed in [19] has been selected in view of the facts that no burdensome minmax optimization problem is required to be solved on-line, and that it naturally provides the future reference trajectories, as it will be clarified later in this chapter. Similarly to [19], a nominal model of subsystem $\mathscr{S}_i$ is associated to equation (3)

$$\hat{\mathbf{x}}_{k+1}^{[i]} = \mathbf{A}_{ii}\hat{\mathbf{x}}_k^{[i]} + \mathbf{B}_{ii}\hat{\mathbf{u}}_k^{[i]} + \sum_{j\in\mathscr{N}_i} \{\mathbf{A}_{ij}\tilde{\mathbf{x}}_k^j + \mathbf{B}_{ij}\tilde{\mathbf{u}}_k^j\} \quad (4)$$

while the control law to be used for $\mathscr{S}_i$ is

$$\mathbf{u}_k^{[i]} = \hat{\mathbf{u}}_k^{[i]} + \mathbf{K}_i(\mathbf{x}_k^{[i]} - \hat{\mathbf{x}}_k^{[i]}) \quad (5)$$

where $\mathbf{K}_i$ must be chosen to satisfy Assumption 1.

Letting $\mathbf{z}_k^{[i]} = \mathbf{x}_k^{[i]} - \hat{\mathbf{x}}_k^{[i]}$, in view of (3), (4), and (5) one has

$$\mathbf{z}_{k+1}^{[i]} = \mathbf{F}_{ii}\mathbf{z}_k^{[i]} + \mathbf{w}_k^{[i]} \tag{6}$$

where $\mathbf{w}_k^{[i]} \in \mathscr{W}_i$. Since $\mathscr{W}_i$ is bounded and $\mathbf{F}_{ii}$ is Schur, there exists a robust positively invariant (RPI) set $\mathscr{Z}_i$ for (6) such that, for all $\mathbf{z}_k^{[i]} \in \mathscr{Z}_i$, then $\mathbf{z}_{k+1}^{[i]} \in \mathscr{Z}_i$. Given $\mathscr{Z}_i$ define, if possible, two sets, neighborhoods of the origin, $\Delta\mathscr{E}_i$ and $\Delta\mathscr{U}_i$, $i = 1,\dots,M$ such that $\Delta\mathscr{E}_i \oplus \mathscr{Z}_i \subseteq \mathscr{E}_i$ and $\Delta\mathscr{U}_i \oplus K_i\mathscr{Z}_i \subseteq \mathscr{E}_i^{\mathscr{U}}$, respectively.

3.1 The online phase: the i-DPC optimization problems

At any time instant k each subsystem $\mathscr{S}_i$ solves the following i-DPC problem.

$$\min_{\hat{\mathbf{x}}_k^{[i]}, \hat{\mathbf{u}}_{[k:k+N-1]}^{[i]}} V_i^N = \sum_{\nu=0}^{N-1} (\|\hat{\mathbf{x}}_{k+\nu}^{[i]}\|_{\mathbf{Q}_i^o}^2 + \|\hat{\mathbf{u}}_{k+\nu}^{[i]}\|_{\mathbf{R}_i^o}^2) + \|\hat{\mathbf{x}}_{k+N}^{[i]}\|_{\mathbf{P}_i^o}^2 \tag{7}$$

subject to (4),

$$\mathbf{x}_k^{[i]} - \hat{\mathbf{x}}_k^{[i]} \in \mathscr{Z}_i \tag{8}$$

and, for $\nu = 0,\dots,N-1$

$$\hat{\mathbf{x}}_{k+\nu}^{[i]} - \tilde{\mathbf{x}}_{k+\nu}^{[i]} \in \Delta\mathscr{E}_i \tag{9}$$

$$\hat{\mathbf{u}}_{k+\nu}^{[i]} - \tilde{\mathbf{u}}_{k+\nu}^{[i]} \in \Delta\mathscr{U}_i \tag{10}$$

$$\hat{\mathbf{x}}_{k+\nu}^{[i]} \in \hat{\mathscr{X}}_i \subseteq \mathscr{X}_i \ominus \mathscr{Z}_i \tag{11}$$

$$\hat{\mathbf{u}}_{k+\nu}^{[i]} \in \hat{\mathscr{U}}_i \subseteq \mathscr{U}_i \ominus \mathbf{K}_i\mathscr{Z}_i \tag{12}$$

and to the terminal constraint

$$\hat{\mathbf{x}}_{k+N}^{[i]} \in \hat{\mathscr{X}}_i^F \tag{13}$$

The choice of the positive definite matrices $\mathbf{Q}_i^o$, $\mathbf{R}_i^o$, and $\mathbf{P}_i^o$ in (7) is discussed in [15,6] to guarantee stability and convergence, while $\hat{\mathscr{X}}_i^F$ in (13) is a nominal terminal set which must be chosen to satisfy the following assumption.

Assumption 2 *Letting* $\hat{\mathscr{X}} = \prod_{i=1}^M \hat{\mathscr{X}}_i$, $\hat{\mathscr{U}} = \prod_{i=1}^M \hat{\mathscr{U}}_i$ *and* $\hat{\mathscr{X}}^F = \prod_{i=1}^M \hat{\mathscr{X}}_i^F$, *it holds that:*

(i) $\hat{\mathscr{X}}^F \subseteq \hat{\mathscr{X}}$ *is an invariant set for* $\hat{\mathbf{x}}_{k+1} = (\mathbf{A}+\mathbf{B}\mathbf{K})\hat{\mathbf{x}}_k$;
(ii) $\hat{\mathbf{u}} = \mathbf{K}\hat{\mathbf{x}} \in \hat{\mathscr{U}}$ *for any* $\hat{\mathbf{x}} \in \hat{\mathscr{X}}^F$;
(iii) for all $\hat{\mathbf{x}}_\mathbf{k} \in \hat{\mathscr{X}}^F$ *and, for a given constant* $\kappa > 0$,

$$\mathbf{V}^F(\hat{\mathbf{x}}_{k+1}) - \mathbf{V}^F(\hat{\mathbf{x}}_k) \le -(1+\kappa)\ell(\hat{\mathbf{x}}_k, \mathbf{K}\hat{\mathbf{x}}_k) \tag{14}$$

where $\mathbf{V}^F(\hat{\mathbf{x}}) = \sum_{i=1}^M V_i^F(\hat{\mathbf{x}}^{[i]}) = \sum_{i=1}^M \|\hat{\mathbf{x}}^{[i]}\|_{\mathbf{P}_i^o}^2$ *and* $\ell(\hat{\mathbf{x}}, \hat{\mathbf{u}}) = \sum_{i=1}^M \ell_i(\hat{\mathbf{x}}^{[i]}, \hat{\mathbf{u}}^{[i]}) = \sum_{i=1}^M (\|\hat{\mathbf{x}}^{[i]}\|_{\mathbf{Q}_i^o}^2 + \|\hat{\mathbf{u}}^{[i]}\|_{\mathbf{R}_i^o}^2)$.

At time k, let the pair $\hat{\mathbf{x}}_{k|k}^{[i]}, \hat{\mathbf{u}}_{[k:k+N-1|k]}^{[i]}$ be the solution to the i-DPC problem and define by $\hat{\mathbf{u}}_{k|k}^{[i]}$ the input to the nominal system (4). Then, according to (5), the input to the subsystem (1) is

$$\mathbf{u}_k^{[i]} = \hat{\mathbf{u}}_{k|k}^{[i]} + \mathbf{K}_i(\mathbf{x}_k^{[i]} - \hat{\mathbf{x}}_{k|k}^{[i]}) \tag{15}$$

Denoting by $\hat{\mathbf{x}}_{k+\nu|k}^{[i]}$ the state trajectory of system (4) stemming from $\hat{\mathbf{x}}_{k|k}^{[i]}$ and $\hat{\mathbf{u}}_{[k:k+N-1|k]}^{[i]}$, at time k it is also possible to compute $\hat{\mathbf{x}}_{[k+N|k]}^{[i]}$ and $\mathbf{K}_i\hat{\mathbf{x}}_{k+N|k}^{[i]}$. In DPC, these values incrementally define the trajectories of the reference state and input variables to be used at the next time instant $k+1$, that is

$$\tilde{\mathbf{x}}_{k+N}^{[i]} = \hat{\mathbf{x}}_{k+N|k}^{[i]}, \ \tilde{\mathbf{u}}_{k+N}^{[i]} = \mathbf{K}_i\hat{\mathbf{x}}_{k+N|k}^{[i]} \tag{16}$$

We underline that, in nominal operating conditions, the only information to be transmitted consists in the reference trajectories updated as in (16). More specifically, at time step k, subsystem $\mathscr{S}_i$ computes $\tilde{\mathbf{x}}_{k+N}^{[i]}$ and $\tilde{\mathbf{u}}_{k+N}^{[i]}$ according to (16) and transmits their values to all the subsystems having $\mathscr{S}_i$ as neighbor, allowing them to update the reference trajectories.

3.2 Offline design and initialization

The design of the DPC algorithm requires that a number of tuning parameters are properly selected off-line, i.e., the gain matrices $\mathbf{K}_i$ satisfying Assumption 1, the sets $\mathscr{E}_i$, $\mathscr{E}_i^{\mathscr{U}}$, $\Delta\mathscr{E}_i$, $\Delta\mathscr{U}_i$, and $\mathscr{Z}_i$, and the weighting matrices $\mathbf{Q}_i^o$, $\mathbf{R}_i^o$, and $\mathbf{P}_i^o$, satisfying Assumption 2. In [15,5,6] we have provided solutions to the mentioned realization and design issues.

On the other hand, the initial reference trajectories are also critical tuning parameters, since they strongly affect the initial feasibility. In fact, the initial reference trajectories must be defined off-line, based on the system initial conditions. Moreover, when disturbances of unexpected entity occur during the ordinary system operation, alterating the system's condition, new suitable state and output reference trajectories for all subsystems must be recalculated. Otherwise, possible serious consequences on the future solution (e.g., concerning feasibility) of the control problems could occur. The simplest solution consists (consistently with the approach suggested in [8]) in generating such trajectories using a centralized controller. This has the drawback that a centralized controller must be designed together with the distributed ones, and that it must be kept activated while the system is running in order to recover the proper functioning of the process in case of unpredicted external disturbances. Obviously, this need of a centralized "hidden" supervisor greatly reduces the advantages of utilizing a distributed control scheme.
In [6] two different trajectory generation methods are presented. They can be applied both for offline reference trajectory generation (i.e., performed at time $k = 0$) and for extra-ordinary reset operations, requiring number of iterative information exchanges between neighbors.

4 Properties of DPC

In this section the main theoretical properties of DPC will be discussed, together with some extensions. In order to enhance readability and clarity, the theoretical details will be omitted. The interested reader can rely on [15,14,3] for a more rigorous treatment of the discussed topics.

Control of continuous-time systems and discretization. System (2) can be seen as the state-space representation of a discrete-time empirical model obtained from data through identification procedures, for instance by means of impulse or step response experiments, or it can be computed as the the linearization and discretization of a continuous-time first principle model. In the latter case, the discretization procedure must guarantee to maintain the sparsity of the original continuous-time model, i.e., the mutual influences among the subsystems. In fact, the sparse structure of the model clearly represents physical connections (such as mass or energy flows) between the subsystems.
Unfortunately, the sparsity pattern of the system is lost when the exact ZOH (Zero-Order-Hold), Backward Euler, or bilinear transformations are used, while it is preserved by the Forward Euler (FE) transformation. However, it is well known that with FE some important properties of the underlying continuous time system can be lost; for example stability is maintained only for very small sampling times, which can be inadvisable in many digital control applications.
In distributed and decentralized control techniques based on MPC, where discrete-time models are mainly utilized, the loss of sparsity can easily result in an increase of the controller complexity. For these reasons, in order to improve the performance of FE and to maintain sparsity, a new discretization method called Mixed Euler ZOH (mE-ZOH) has been proposed and analyzed in [12].

Optimality issues and convergence. Global optimality of the interconnected closed loop system cannot be guaranteed using DPC. This is due to the inherent conservativeness of robust algorithms and can be understood in the light of the game-theoretic interpretation of DPC. Namely, the provided solution to the control problem can be cast as a maxmin solution of a dynamic non-cooperative game (see, e.g., [26]) where all the involved subsystems aim to optimize local cost functions which are different from each other: therefore, different and possibly conflicting goals inevitably imply suboptimality.
However, differently from suboptimal distributed MPC algorithms discussed in [24], whose solutions can be regarded as Nash solutions of non-cooperative games and which possibly lead to instability of the closed-loop system, the convergence of the DPC algorithm can be guaranteed. Specifically, in [15] a convergence analysis of the DPC algorithm is carried out, showing that such fundamental property is guaranteed provided that the tuning parameters and the required sets can be chosen as specified and that the feasibility of the i-DPC problems holds at time step $k = 0$.

Output feedback. The DPC approach, described for coping with unknown exogenous additive disturbances, has been employed in [14] for designing a DPC algorithm for output feedback control. Specifically, assume that the input and output equations of the system are the following

$$\begin{aligned} \mathbf{x}_{k+1}^{o[i]} &= \mathbf{A}_{ii}\mathbf{x}_k^{o[i]} + \mathbf{B}_{ii}\mathbf{u}_k^{[i]} + \textstyle\sum_{j=1,j\neq i}^{M}\{\mathbf{A}_{ij}\mathbf{x}_k^{o[j]} + \mathbf{B}_{ij}\mathbf{u}_k^{[j]}\} \\ \mathbf{y}_k^{[i]} &= \mathbf{C}_i\mathbf{x}_k^{o[i]} \end{aligned} \tag{17}$$

where the state which is not directly available is here denoted as $\mathbf{x}^{o[i]}$. Denote by $\mathbf{x}_i$ the estimate of $\mathbf{x}^{o[i]}$, for all $i = 1,\ldots,M$, provided by a decentralized Luenberger-like observer of the type

$$\begin{aligned} \mathbf{x}_{k+1}^{[i]} &= \mathbf{A}_{ii}\mathbf{x}_k^{[i]} + \mathbf{B}_{ii}\mathbf{u}_k^{[i]} + \textstyle\sum_{j=1,j\neq i}^{M}\{\mathbf{A}_{ij}\mathbf{x}_k^{j} + \mathbf{B}_{ij}\mathbf{u}_k^{j}\} \\ &\quad -\mathbf{L}_i(\mathbf{y}_k^{[i]} - \mathbf{C}_i\mathbf{x}_k^{[i]}) \end{aligned} \tag{18}$$

Assume that the decentralized observer is convergent i.e., $\mathbf{A}+\mathbf{LC}$ is Schur, where $\mathbf{C}=\text{diag}(\mathbf{C}_1,\ldots,\mathbf{C}_M)$ and $\mathbf{L}=\text{diag}(\mathbf{L}_1,\ldots,\mathbf{L}_M)$. Under this assumption it is possible to guarantee that the estimation error for each subsystem is bounded, i.e., $\mathbf{x}_k^{o[i]} - \mathbf{x}_k^{[i]} \in \Sigma_i$ for all $i = 1,\ldots,M$. In this way (18) exactly corresponds with the perturbed system (1), where $\mathbf{d}_k^{[i]} = -\mathbf{L}_i(\mathbf{y}_k^{[i]} - \mathbf{C}_i\mathbf{x}_k^{[i]})$ is regarded as a bounded disturbance, i.e., $\mathbf{d}_k^{[i]} \in \mathscr{D}_i = -\mathbf{L}_i\mathbf{C}_i\Sigma_i$. From this point on, the output feedback control problem is solved as a robust state feedback problem applied to the system (1). Details on this approach can be found in [14], where a condition and a constructive method are derived to compute the sets Σ_i in such a way that $\Sigma = \prod_{i=1}^{M}\Sigma_i$ is an invariant set for the interconnected observer error.

Tracking. To extend the DPC method for tracking desired output signals, the main problem is to characterize the state and input trajectories, for all subsystems, which correspond to the desired output trajectories. To clarify this, consider the case where the desired output trajectories are constant and equal to $\bar{\mathbf{y}}^{[i]}$, for all $i = 1,\ldots,M$. Under standard assumptions on the system matrices $(\mathbf{A},\mathbf{B},\mathbf{C})$ (i.e., that the input/output collective system obtained from (17) has no invariant zeros in 1, see [3] for details), the desired setpoint values for the collective state $\mathbf{x}^o$ and input $\mathbf{u}$ can be computed as follows

$$\begin{bmatrix} \bar{\mathbf{x}}^o \\ \bar{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}-\mathbf{I} & \mathbf{B} \\ \mathbf{C} & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \bar{\mathbf{y}} \end{bmatrix} \tag{19}$$

where $\bar{\mathbf{y}} = (\bar{\mathbf{y}}^{[1]}\ldots\bar{\mathbf{y}}^{[1]})$. In this way, the setpoint values $\bar{\mathbf{x}}^{[i]}$ and $\bar{\mathbf{u}}^{[i]}$ for $\mathbf{x}^{[i]}$ and $\mathbf{u}^{[i]}$, respectively, are obtained as the vector components of $\bar{\mathbf{x}}^o$ and $\bar{\mathbf{u}}$ of suitable dimensions.
However, the solution to (19) requires either a centralized

computation or an iterative procedure (to be carried out within a sampling interval). These solutions are not compatible with the proposed approach. Two different solutions have been proposed to circumvent this problem: (i) the use of the so-called *velocity form*, see [3]; (ii) the introduction of a multi-layer control architecture, see [11,10].

i) The use of the velocity form (as discussed in [3]) implies a transformation of (17) into an equivalent system, whose state variable is the pair $(\delta\mathbf{x}^{[i]}, \boldsymbol{\varepsilon}^{[i]})$ and whose input variable is $\delta\mathbf{u}^{[i]}$, where $\delta\mathbf{x}_k^{[i]} = \mathbf{x}_k^{[i]} - \mathbf{x}_{k-1}^{[i]}$, $\boldsymbol{\varepsilon}_i = \mathbf{y}_k^{[i]} - \overline{\mathbf{y}}^{[i]}$, and $\delta\mathbf{u}_k^{[i]} = \mathbf{u}_k^{[i]} - \mathbf{u}_{k-1}^{[i]}$. In this way, the tracking problem for (17) is cast as a more standard regulation problem for the velocity form without having to explicitly compute $\overline{\mathbf{x}}^{[i]}$ and $\overline{\mathbf{u}}^{[i]}$. Therefore, the DPC algorithm can be applied without significant restrictions and has also the advantage of guaranteeing offset-free tracking in presence of constant perturbations.
The main issue in this solution consists of the analysis on how constraints on $\mathbf{x}^{[i]}$ and on $\mathbf{u}^{[i]}$ translate into constraints on $\delta\mathbf{x}^{[i]}$, $\boldsymbol{\varepsilon}^{[i]}$, and $\delta\mathbf{u}^{[i]}$, especially as far as the terminal constraints are concerned.

ii) An alternative solution is discussed in [11,10], where a hierarchical control architecture is proposed, see Figure 1: a *reference output trajectory layer* computes in a distributed way the output reference trajectories $\tilde{\mathbf{y}}^{[i]}$ given the "ideal" set-points $\overline{\mathbf{y}}^{[i]}$, while a *reference state and input trajectory layer* determines the corresponding state and control trajectories $\tilde{\mathbf{x}}^{[i]}$ and $\tilde{\mathbf{u}}^{[i]}$. At the lower layer of the structure of Figure 1, a distributed *robust MPC layer* is designed to drive the real state and input trajectories $\mathbf{x}_k^{[i]}$ and $\mathbf{u}_k^{[i]}$ of the subsystems as close as possible to $\tilde{\mathbf{x}}_k^{[i]}$, $\tilde{\mathbf{u}}_k^{[i]}$, while satisfying the constraints. Concerning the *reference output trajectory layer*, it is important to remark that, in the considered distributed context, too rapid changes of the output reference trajectory of a given subsystem could greatly affect the performance and the behavior of the other subsystems. Therefore, the rate of variation of $\tilde{\mathbf{y}}_k^{[i]}$ is limited, which may limit the reactivity of the proposed control scheme to rapid changes in the output setpoints.
The main advantages of the scheme proposed in [10] are: scalability of the online implementation, limited transmission and computational load (also in view of the facts that the reference generator layer is independent of the robust MPC layer, and hence computations can be performed in a parallelized fashion, and that information is required to be transmitted only among neighboring subsystems), and simplicity of implementation.

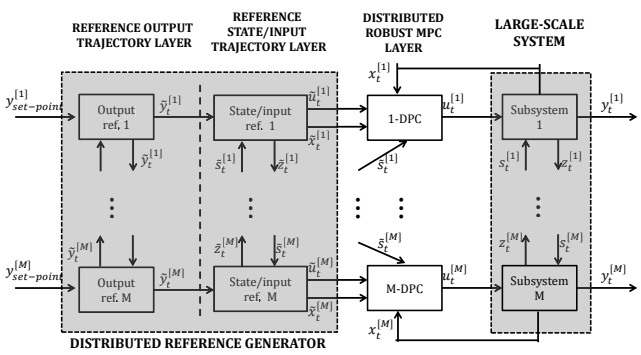

Fig. 1. Distributed architecture for tracking reference signals.

5 Applications of DPC

The DPC algorithm has been tested in a number of different test cases, see e.g.,[15,5,6,10]. In this section we first show some results of application of DPC to three realistic simulation examples (see Sections 5.1-5.3), previously shown in [6] and related to popular case studies in the context of distributed control. Also, we apply the multi-layer DPC scheme for tracking to a real test bed, i.e., the control of a small fleet of unicycle robots (see Section 5.4), previously illustrated in [10].

5.1 Temperature control

We aim at regulating the temperatures T_A, T_B, T_C and T_D of the four rooms of the building represented in Figure 2 (see [11,4]). The first apartment is constituted by rooms A and B, while the second one by rooms C and D. Each room is equipped with a radiator supplying heats q_A, q_B, q_C and q_D. The heat transfer coefficient between rooms A - C and B - D is $k_1^t = 1$ W/m^{2}K, the one between rooms A - B and C - D is $k_2^t = 2.5$ W/m^{2}K, and the one between each room and the external environment is $k_e^t = 0.5$ W/m^{2}K. The nominal external temperature is $\overline{T}_E = 0\,^\circ$C and, for the sake of simplicity, solar radiation is not considered. The volume of each room is $V = 48$ m^3, and the wall surfaces between the rooms are all equal to $s_r = 12$ m^2, while those of the external walls are equal to $s_e = 24$ m^2. Air density and heat capacity are $\rho = 1.225$ Kg/m^3 and $c = 1005$ J/KgK, respectively. Letting $\phi = \rho cV$, the dynamic model is the following:

$$\begin{aligned}
\phi\,\tfrac{dT_A}{dt} &= s_r k_2^t(T_B - T_A) + s_r k_1^t(T_C - T_A) + s_e k_e^t(T_E - T_A) + q_A\\
\phi\,\tfrac{dT_B}{dt} &= s_r k_2^t(T_A - T_B) + s_r k_1^t(T_D - T_B) + s_e k_e^t(T_E - T_B) + q_B\\
\phi\,\tfrac{dT_C}{dt} &= s_r k_1^t(T_A - T_C) + s_r k_2^t(T_D - T_C) + s_e k_e^t(T_E - T_C) + q_C\\
\phi\,\tfrac{dT_D}{dt} &= s_r k_1^t(T_B - T_D) + s_r k_2^t(T_C - T_D) + s_e k_e^t(T_E - T_D) + q_D
\end{aligned}$$

The considered equilibrium point is: $q_A = q_B = q_C = q_D = \overline{q} = \overline{T} s_e k_e^t$, with $T_A = T_B = T_C = T_D = \overline{T} = 20\,^\circ$C. Let $\delta T_A = T_A - \overline{T}$, $\delta T_B = T_B - \overline{T}$, $\delta T_C = T_C - \overline{T}$, $\delta T_D = T_D - \overline{T}$, $\delta T_E = T_E - \overline{T}_E$, $\delta q_A = (q_A - \overline{q})/c\rho V$, $\delta q_B = (q_B - \overline{q})/c\rho V$, $\delta q_C = (q_C - \overline{q})/c\rho V$ and $\delta q_D = (q_D - \overline{q})/c\rho V$. In this way, denoting $\sigma_1 = s_r k_1^t/c\rho V$, $\sigma_2 = s_r k_2^t/c\rho V$, $\sigma_3 = s_e k_e^t/c\rho V$, $\sigma = \sigma_1 + \sigma_2 + \sigma_3$, $\mathbf{x} = (\delta T_A, \delta T_B, \delta T_C, \delta T_D)$, $\mathbf{u} = (\delta q_A, \delta q_B, \delta q_C, \delta q_D)$ and $\mathbf{d} = [\sigma_e\ \sigma_e\ \sigma_e\ \sigma_e]^T \delta T_E$ the previous model is rewritten in state space representation

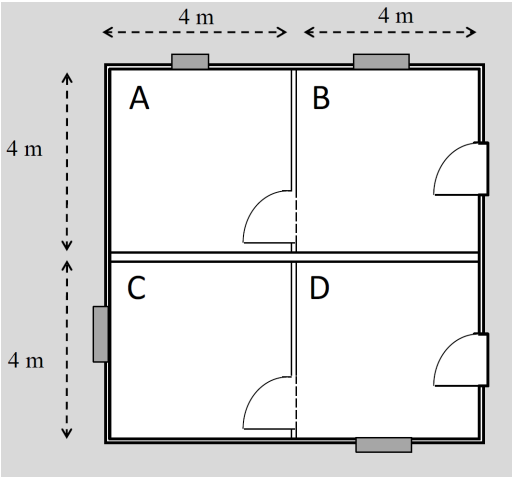

Fig. 2. Schematic representation of a building with two apartments.

$\dot{\mathbf{x}}(t) = \mathbf{A_c}\,\mathbf{x}(t) + \mathbf{B_c}\mathbf{u}(t) + \mathbf{d}(t)$, where

$$\mathbf{A_c} = \begin{bmatrix} -\sigma & \sigma_2 & \sigma_1 & 0 \\ \sigma_2 & -\sigma & 0 & \sigma_1 \\ \sigma_1 & 0 & -\sigma & \sigma_2 \\ 0 & \sigma_1 & \sigma_2 & -\sigma \end{bmatrix}, \mathbf{B_c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The discrete-time system of the form (2) (with $n = 4$ and $m = 4$) is obtained by mE-ZOH discretization [12] with sampling time $h = 10$. The partition of inputs and states is:

$$x^{[1]} = \begin{bmatrix} \delta T_A & \delta T_B \end{bmatrix}^T, \; u^{[1]} = \begin{bmatrix} \delta q_A & \delta q_B \end{bmatrix}^T$$

$$x^{[2]} = \begin{bmatrix} \delta T_C & \delta T_D \end{bmatrix}^T, \; u^{[2]} = \begin{bmatrix} \delta q_C & \delta q_D \end{bmatrix}^T$$

The constraints on the inputs and the states of the linearized system have been chosen as:

$$x^{[1]}_{min} = \begin{bmatrix} -5 & -5 \end{bmatrix}^T, \; x^{[1]}_{max} = \begin{bmatrix} 5 & 5 \end{bmatrix}^T$$

$$x^{[2]}_{min} = \begin{bmatrix} -5 & -5 \end{bmatrix}^T, \; x^{[2]}_{max} = \begin{bmatrix} 5 & 5 \end{bmatrix}^T$$

$$u^{[1]}_{min} = \begin{bmatrix} -0.038 & -0.038 \end{bmatrix}^T, \; u^{[1]}_{max} = \begin{bmatrix} 0.030 & 0.030 \end{bmatrix}^T$$

$$u^{[2]}_{min} = \begin{bmatrix} -0.038 & -0.038 \end{bmatrix}^T, \; u^{[2]}_{max} = \begin{bmatrix} 0.030 & 0.030 \end{bmatrix}^T$$

For implementation details, see [6]. In the simulations reported below, the perturbed initial conditions for $\delta T_A = -3.2\,^\circ$C, $\delta T_B = -2.58\,^\circ$C, $\delta T_C = -1.12\,^\circ$C, $\delta T_D = 3.55\,^\circ$C have been set, the real external temperature has been assumed to randomly vary between $-10\,^\circ$C and $10\,^\circ$C and a sudden decrease of temperature T_A has been forced at $t = 350s$, representing for instance the opening of a door, to show the capability of DPC to recover the reference trajectories.

The results of the simulations, performed using the continuous-time process model, are shown in Figure 3, while the values of the input variables are depicted in Figure 4. In both these figures a comparison between DPC and a centralized MPC (cMPC), with the same state and control weighting matrices is provided, showing only a small reduction of performances.

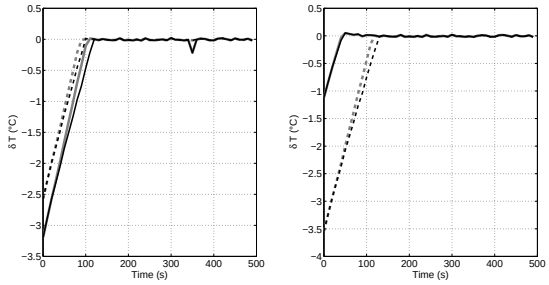

Fig. 3. State trajectories with DPC (black lines) and cMPC (gray lines) of δT_A (left, solid lines), δT_B (left, dashed lines), δT_C (right, solid lines), δT_D (right, dashed lines).

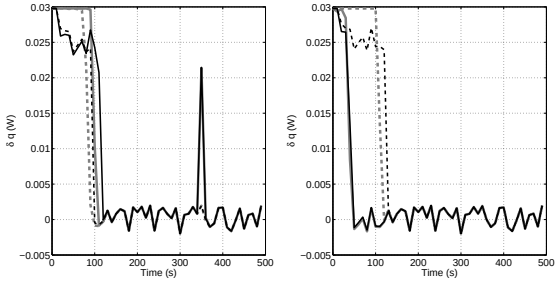

Fig. 4. Input trajectories with DPC (black lines) and cMPC (gray lines) of δq_A (left, solid lines), δq_B (left, dashed lines), δq_C (right, solid lines), δq_D (right, dashed lines).

To quantitatively assess the performance deterioration of DPC with respect to cMPC, the following two indices have been considered

$$ISRE = \sum_{i=1}^{M} \int_0^{T_{end}} \sqrt{x^{[i]}(t)^T x(t)^{[i]}}\, dt \tag{20}$$

$$J = \sum_{i=1}^{M} \sum_{k=0}^{N_{end}} x_k^{[i]T} Q_i x_k^{[i]} + u_k^{[i]\prime} R_i u_k^{[i]} \tag{21}$$

where T_{end} is the final time and N_{end} is the total number of discrete-time steps of the simulation experiment. The values of *ISRE* and J corresponding to the state transients of Figure 3 and Figure 4 with DPC and cMPC are reported in Table 5.1.

5.2 Four-tanks system

A benchmark case often used to assess the effectiveness of distributed control algorithms is the four-tanks system schematically drawn in Figure 5, originally described in [16]

ISRE	cMPC	461.4
	DPC	501.1
	DPC/cMPC	1.09
J	cMPC	120.2
	DPC	127.8
	DPC/cMPC	1.06

Table 1
ISRE and *J* with DPC and cMPC in the temperature control problem.

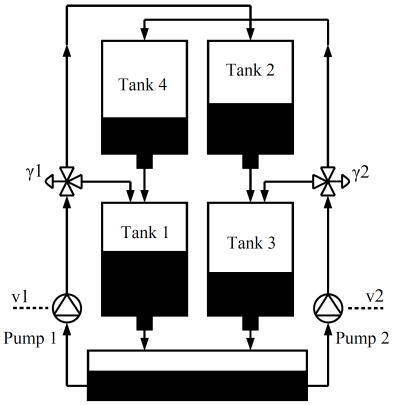

Fig. 5. Schematic representation of a four-tanks system.

and then utilized, for instance, in [1,20,3]. The goal is to regulate the levels h_1, h_2, h_3 and h_4 of the four tanks. The manipulated inputs are the voltages of the two pumps v_1 and v_2. We assume to have a bounded unknown disturbance $w = (w_1, w_2)$ on the applied voltages, such that the real input to the plant is $(v_1 + w_1, v_2 + w_2)$. Let the parameters γ_1 and $\gamma_2 \in (0,1)$ represent the fraction of water that flows inside the lower tanks, and are kept fixed during the simulations. Then, the dynamics of the system is given by

$$\begin{aligned}
\frac{dh_1}{dt} &= -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_4}{A_4}\sqrt{2gh_4} + \frac{\gamma_1 k_1}{A_1}v_1 \\
\frac{dh_2}{dt} &= -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{(1-\gamma_1)k_1}{A_2}v_1 \\
\frac{dh_3}{dt} &= -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{a_2}{A_2}\sqrt{2gh_2} + \frac{\gamma_2 k_2}{A_3}v_2 \\
\frac{dh_4}{dt} &= -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_2)k_2}{A_4}v_2
\end{aligned} \tag{22}$$

where A_i and a_i are the cross-section of Tank i and the cross section of the outlet hole of Tank i, respectively. The coefficients k_1 and k_2 represent the conversion parameters from the voltage applied to the pump to the flux of water. The values of the parameters, taken from [16], are: $A_1 = A_4 = 28$ cm^2, $A_2 = A_3 = 32$ cm^2, $a_1 = a_4 = 0.071$ cm^2, $a_2 = a_3 = 0.057$ cm^2, $k_1 = 3.35$ cm$^3/Vs$, $k_2 = 3.33$ cm$^3/Vs$, $\gamma_1 = 0.7$, $\gamma_2 = 0.6$. The considered equilibrium point is $\bar{v}_1 = \bar{v}_2 = \underline{3}$ V, $h_1 = 12.263$ cm, $h_2 = 1.409$ cm, $\underline{h}_3 = 12.783$ cm and $h_4 = 1.634$ cm. Denoting $\delta h_l = h_l - \underline{h}_l$, $l = 1,2,3,4$ and $\delta v_i = v_i - \bar{v}_i$, $i = 1,2$, $\mathbf{x} = (\delta h_1, \delta h_2, \delta h_3, \delta h_4)$, $\mathbf{u} = (\delta v_1, \delta v_2)$, $\mathbf{d} = \mathbf{B}(w_1, w_2)$, linearizing system (22) around the considered equilibrium point and discretizing it using mE-ZOH [12] with sampling time $h = 1$ s, we obtain a linear system of the type (2), where

$$\mathbf{A} = \begin{bmatrix} 0.98 & 0 & 0 & 0.04 \\ 0 & 0.97 & 0 & 0 \\ 0 & 0.03 & 0.99 & 0 \\ 0 & 0 & 0 & 0.96 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0.08 & 0 \\ 0.03 & 0 \\ 0 & 0.06 \\ 0 & 0.05 \end{bmatrix}$$

The inputs and states are partitioned as:

$$x^{[1]} = \begin{bmatrix} \delta h_1 & \delta h_2 \end{bmatrix}^T, \; u^{[1]} = \delta v_1$$

$$x^{[2]} = \begin{bmatrix} \delta h_3 & \delta h_4 \end{bmatrix}^T, \; u^{[2]} = \delta v_2$$

The constraints on the inputs and the states of the linearized system have been chosen as:

$$x^{[1]}_{min} = \begin{bmatrix} -12.263 & -1.409 \end{bmatrix}^T, \; x^{[1]}_{max} = \begin{bmatrix} 40 & 40 \end{bmatrix}^T + x^{[1]}_{min}$$

$$x^{[2]}_{min} = \begin{bmatrix} -12.783 & -1.634 \end{bmatrix}^T, \; x^{[2]}_{max} = \begin{bmatrix} 40 & 40 \end{bmatrix}^T + x^{[2]}_{min}$$

$$u^{[1]}_{min} = u^{[2]}_{min} - 3, \; u^{[1]}_{max} = u^{[2]}_{max} = 3$$

The disturbances $w_{1,2}$ on the applied voltages are assumed to randomly vary between -0.01 V and 0.01 V. For implementation details see [6]. Starting from initial conditions

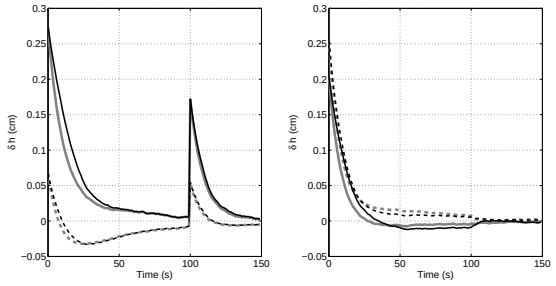

Fig. 6. Trajectories of the states $x^{[1]}$ (left) and $x^{[2]}$ (right) obtained with DPC (black lines) and with cMPC (gray lines) for the four–tanks system. Solid lines: δh_1 and δh_3; dashed lines: δh_2 and δh_4.

$\delta h_1 = 0.274$ cm, $\delta h_2 = 0.067$ cm, $\delta h_3 = 0.203$ cm, and $\delta h_4 = 0.254$ cm. The simulation results, obtained using the continuous-time nonlinear model, are reported in Figure 6, while in Figure 7 the applied real voltages are shown. In addition to the external disturbance (w_1, w_2), included in the robust controller design, at time $t = 100$ s an unpredicted impulse equal to 2 V has been applied to the first pump. The reference trajectories were then re-generated online to recover the nominal operating conditions. The performances are close to the ones obtained with centralized MPC, as also witnessed by the values taken by the indices *ISRE* and *J* defined in (20), (21) and reported in Table 5.2.

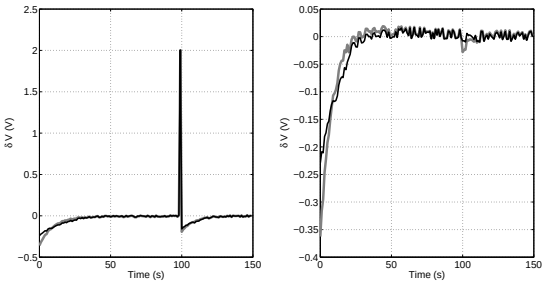

Fig. 7. Inputs δv_1 (left) and δv_2 (right) obtained with DPC (black lines) and with cMPC (gray lines) for the four-tanks system.

ISRE	cMPC	74.2
	DPC	82.3
	DPC/cMPC	1.11
J	cMPC	1.36
	DPC	1.47
	DPC/cMPC	1.08

Table 2
ISRE and *J* with DPC and cMPC in the in the four-tanks system problem.

5.3 Cascade coupled flotation tanks

The third example deals with the level control problem of flotation tanks proposed in [28]. The system is constituted by five tanks connected in cascade with control valves between the tanks (Figure 8). A flow of pulp q enters the first tank. The goal is to keep stable the levels y_i, $i = 1,\dots,5$, in all the tanks. The manipulated inputs are the commands to the valves v_i, $i = 1,\dots,5$. The mathematical model describing

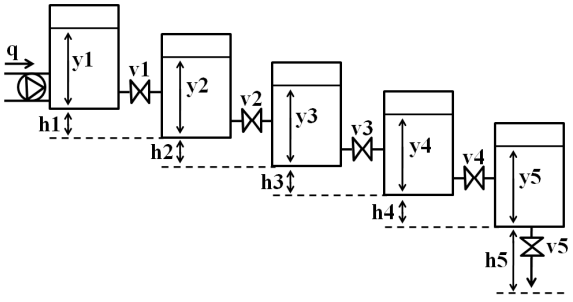

Fig. 8. Schematic representation of the flotation tanks.

the dynamics of the levels inside the five tanks is [28]:

$$\begin{aligned}
\pi r^2 \tfrac{dy_1}{dt} &= q - k_1 v_1 \sqrt{y_1 - y_2 + h_1} \\
\pi r^2 \tfrac{dy_2}{dt} &= k_1 v_1 \sqrt{y_1 - y_2 + h_1} - k_2 v_2 \sqrt{y_2 - y_3 + h_2} \\
\pi r^2 \tfrac{dy_3}{dt} &= k_2 v_2 \sqrt{y_2 - y_3 + h_2} - k_3 v_3 \sqrt{y_3 - y_4 + h_3} \\
\pi r^2 \tfrac{dy_4}{dt} &= k_3 v_3 \sqrt{y_3 - y_4 + h_3} - k_4 v_4 \sqrt{y_4 - y_5 + h_4} \\
\pi r^2 \tfrac{dy_5}{dt} &= k_4 v_4 \sqrt{y_4 - y_5 + h_4} - k_5 v_5 \sqrt{y_5 + h_5}
\end{aligned} \tag{23}$$

where r is radius of the tanks, k_i, $i = 1,\dots,5$ are the valves coefficients and h_i, $i = 1,\dots,5$ are the physical height differences between subsequent tanks. We set $r = 1$ *m*, $k_i = 0.1$ $\mathrm{m}^{2.5}$/Vs, $i = 1,\dots,5$ and $h_i = 0.5$ m, $i = 1,\dots,5$. The nominal value for the inlet flow is $\bar{q} = 0.1$ m^3/s and we assume it is affected by an uncertainty $w = \pm 0.5\%$ randomly varying with the time. We considered the equilibrium point where $\bar{y_i} = 2$ m, $i = 1,\dots,5$, and, correspondingly, $\bar{v_i} = 1.4142$ V, $i = 1,\dots,4$ and $\bar{v_5} = 0.6325$ V. Let $\delta y_i = y_i - \bar{y_i}$, $i = 1,\dots,5$, $\delta v_i = v_i - \bar{v_i}$, $i = 1,\dots,5$, $\mathbf{x} = (\delta y_1, \delta y_2, \delta y_3, \delta y_4, \delta y_5)$, $\mathbf{u} = (\delta v_1, \delta v_2, \delta v_3, \delta v_4, \delta v_5)$ and $\mathbf{d} = \mathbf{B}_d w$. The linearization of system (23) in correspondence of the considered equilibrium point and its discretization with mE-ZOH using a sampling time 5 s, leads to a linear system of the form (2) , where $\mathbf{B}_d = [1.4714\ 0\ 0\ 0\ 0]^T$ and

$$\mathbf{A} = \begin{bmatrix} 0.853 & 0.147 & 0 & 0 & 0 \\ 0.136 & 0.727 & 0.136 & 0 & 0 \\ 0 & 0.136 & 0.727 & 0.136 & 0 \\ 0 & 0 & 0.136 & 0.727 & 0.136 \\ 0 & 0 & 0 & 0.157 & 0.969 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} -0.104 & 0 & 0 & 0 & 0 \\ 0.096 & -0.096 & 0 & 0 & 0 \\ 0 & 0.096 & -0.096 & 0 & 0 \\ 0 & 0 & 0.096 & -0.096 & 0 \\ 0 & 0 & 0 & 0.111 & -0.248 \end{bmatrix}$$

The partitions of inputs and states, for $i = 1,\dots,5$ is:

$$x^{[i]} = \delta y_i,\ u^{[1]} = \delta v_1$$

The constraints on the inputs and the states of the linearized system, for $i = 1,\dots,5$, have been set as:

$$x^{[i]}_{min} = -1,\ x^{[i]}_{max} = 1,\ u^{[i]}_{min} = -\bar{v_i},\ u^{[i]}_{max} = 3 - \bar{v_i}$$

For implementation details, please see [6].

The initial levels of the tanks have been assumed to be different from the required values, that is $\delta y_1 = -23.3$ cm, $\delta y_2 = -21.6$ cm, $\delta y_3 = 23.3$ cm, $\delta y_4 = 44.4$ cm, and $\delta y_5 = -12.9$ cm and at time $t = 300s$ a disturbance of magnitude $w = 0.1$ m^3/s has been applied to the plant. In Figure 9 we show the transients, obtained using the continuous-time nonlinear model, of the state and input of the first tank, directly affected by the external flow q. Figure 10 and Figure 11 report, respectively, the states and the inputs of the remaining four tanks. Note that, also in this case, the distributed control system reacts to the disturbance by generating from scratch the reference trajectories. Moreover, only minor differences arise between the centralized and the distributed solutions, as again shown by the indices (20), (21) reported in Table 5.3.

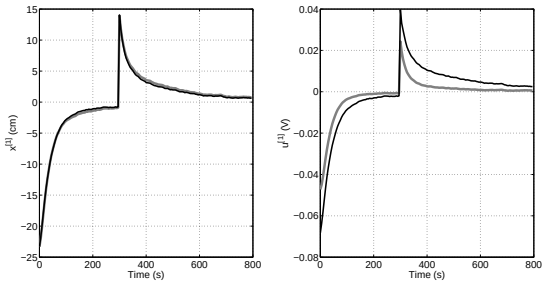

Fig. 9. Trajectories of the state $x^{[1]}$ (left) and of the input $u^{[1]}$ (right) obtained with DPC (black lines) and with cMPC (gray lines) for the control of the floating tanks.

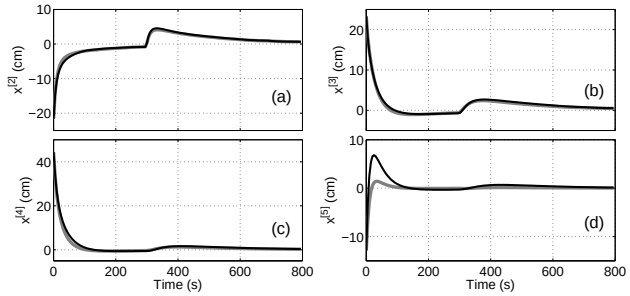

Fig. 10. Trajectories of the states $x^{[2]}$ (a), $x^{[3]}$ (b), $x^{[4]}$ (c) and $x^{[5]}$ (d) obtained with DPC (black lines) and with cMPC (gray lines) for the control of the floating tanks.

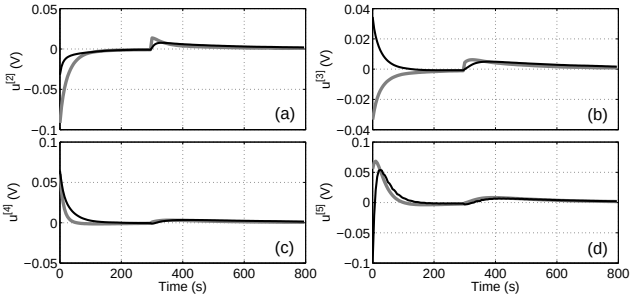

Fig. 11. Inputs $u^{[2]}$ (a), $u^{[3]}$ (b), $u^{[4]}$ (c) and $u^{[5]}$ (d) obtained with DPC (black lines) and with cMPC (gray lines) for the control of the floating tanks.

5.4 Control of unicycle robots

In this section the proposed algorithm is applied to the problem of positioning a number of mobile robots in specified positions, while guaranteeing collision avoidance.
The dynamics of a single robot is described by a modified version of the first-order kinematic model [23]:

$$\dot{x} = v\cos\phi \tag{24a}$$
$$\dot{y} = v\sin\phi \tag{24b}$$
$$\dot{\phi} = \omega \tag{24c}$$
$$\dot{v} = a \tag{24d}$$

ISRE	cMPC	10.7
	DPC	12.2
	DPC/cMPC	1.14
J	cMPC	6.6
	DPC	6.7
	DPC/cMPC	1.01

Table 3
ISRE and *J* with DPC and cMPC in the flotation tanks control problem.

where (x,y) is the cartesian position of the robot, ϕ is its orientation angle, and v is its linear velocity. The linear acceleration a and the angular velocity ω are inputs.
By resorting to a feedback linearization procedure (see [23]) a linear model of the robots can be used to describe the system's dynamics. Namely, define $\eta_1 = x$, $\eta_2 = \dot{x}$, $\eta_3 = y$, $\eta_4 = \dot{y}$, and the dynamics resulting from (24) is

$$\dot{\eta}_1 = \eta_2 \tag{25a}$$
$$\dot{\eta}_2 = a\cos\phi - v\omega\sin\phi \tag{25b}$$
$$\dot{\eta}_3 = \eta_4 \tag{25c}$$
$$\dot{\eta}_4 = a\sin\phi + v\omega\cos\phi \tag{25d}$$

Now define two new "fictitious" input variables $a_x = a\cos\phi - v\omega\sin\phi$ and $a_y = a\sin\phi + v\omega\cos\phi$. From (25) the model (24) is transformed in a set of two decoupled double integrators with inputs a_x and a_y.
To recover the real inputs (ω, a) from (a_x, a_y) compute

$$\begin{bmatrix}\omega\\ a\end{bmatrix} = \frac{1}{v}\begin{bmatrix}-\sin\phi & \cos\phi\\ v\cos\phi & v\sin\phi\end{bmatrix}\begin{bmatrix}a_x\\ a_y\end{bmatrix} \tag{26}$$

Note that, for obtaining (26), it is assumed that $v \neq 0$. This singularity point must be accounted for when designing control laws on the equivalent linear model [23].
In discrete-time, from (25) and with sampling time $\tau = 5$ s, we obtain

$$A_{ii} = A = \begin{bmatrix}1 & \tau & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & \tau\\ 0 & 0 & 0 & 1\end{bmatrix}, B_{ii} = B = \begin{bmatrix}\frac{\tau^2}{2} & 0\\ \tau & 0\\ 0 & \frac{\tau^2}{2}\\ 0 & \tau\end{bmatrix}$$

The measured variables are x and y, i.e., η_1 and η_3 in (25). Note that this case study is characterized by (i) no dynamically coupling terms, i.e., $A_{ij} = 0$ and $B_{ij} = 0$ for all $i, j = 1,\ldots,M$ with $j \neq i$; (ii) static coupling constraints on the position variables guaranteeing collision avoidance.

The experimental set-up consists of three e-puck mobile robots [21]. To simplify the application of the algorithm, the control law is designed on a portable computer communicating with the e-puck robots through wireless connection.

The measurement system consists of a camera, installed on the top of the 130×80 cm^2 working area. Position and orientation of each robot are detected using two colored circles, placed on the top of each agent, see Figure 12.

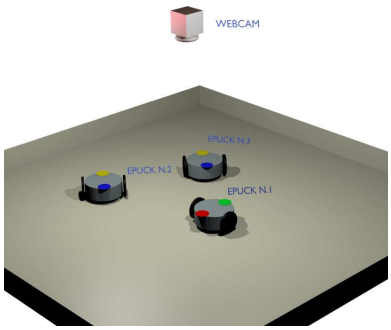

Fig. 12. Sketch of the experimental set-up.

Collision avoidance constraints are in principle non-convex and described using nonlinear inequalities. To circumvent this problem, suitable linear constraints are defined to replace non-convex ones and are obtained by tracing a line stemming from the center of each robot and corresponding to a tangent line to the circumference of the neighboring ones.

In the reported real experiment the three robots are initially placed (at time $t = 1$) at positions $(28,52)$, $(39,16)$, and $(90,39)$ - all coordinates are in cm. Figure 13 shows the evolution of their motion in reaching the goal positions - i.e., $(86,13)$, $(77,55)$, and $(20,39)$ - at time $t = 45$ s while fulfilling collision avoidance constraints.

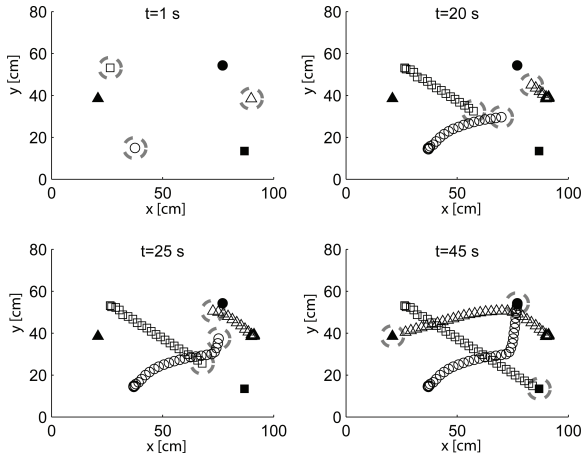

Fig. 13. Plots of the robot trajectories. Robot 1: □; robot 2: ◯; robot 3: △. Symbols with white surface denote the position of the robots, while symbols with black surface denote the goal positions. Large circles with grey dashed line denote the area occupied by the robots.

6 Conclusions

In this paper we have presented the main ideas behind the Distributed Predictive Control (DPC) algorithm first proposed in [15]. In the paper we have also illustrated the main properties, extensions, and applications of DPC, subject of subsequent works.

Acknowledgements

The authors would like to thank Giancarlo Ferrari Trecate and Stefano Riverso for fruitful discussions.

References

[1] I. Alvarado, D. Limon, D. Muñoz de la Peña, J.M. Maestre, M.A. Ridao, H. Scheu, W. Marquardt, R.R. Negenborn, B. De Schutter, F. Valencia, and J. Espinosa. A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark. *Journal of Process Control*, 21(5):800 – 815, 2011.

[2] G. Betti. *Methods and applications of distributed and decentralized model predictive control*. PhD thesis, Politecnico di Milano, Milan, Italy, 2013.

[3] G. Betti, M. Farina, and R. Scattolini. Distributed predictive control for tracking constant references. In *Proceedings of the IEEE American Control Conference (ACC)*, pages 6364–6369, 2012.

[4] G. Betti, M. Farina, and R. Scattolini. Decentralized predictive control for tracking constant references. In *Proceedings of the 52th IEEE Conference on Decision and Control*, pages 5228–5233, 2013.

[5] G. Betti, M. Farina, and R. Scattolini. Distributed predictive control: a non-cooperative approach based on robustness concepts. In R. Negenborn and J. Maestre, editors, *Distributed MPC made easy*. Springer, 2013.

[6] G. Betti, M. Farina, and R. Scattolini. Realization issues, tuning, and testing of a distributed predictive control algorithm. *Journal of Process Control*, 24(4):424 – 434, 2014.

[7] P. D. Christofides, Scattolini R., Muñoz de la Peña D., and Liu J. Distributed model predictive control: a tutorial review and future research directions. *Computers and Chemical Engineering*, 51(4):21–41, 2013.

[8] W. B. Dunbar. Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Transactions on Automatic Control*, 52(7):1249–1263, 2007.

[9] W. B. Dunbar and Murray R. M. Distributed receding horizon control of multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, 2006.

[10] M. Farina, G. Betti, L. Giulioni, and R. Scattolini. An approach to distributed predictive control for tracking - theory and applications. *IEEE Transactions on Control Systems Technology*, 22(4):1558–1566, 2014.

[11] M. Farina, G. Betti, and R. Scattolini. A solution to the tracking problem using distributed predictive control. In *Proceedings of the 12th European Control Conference (ECC), 2013*, pages 4347–4352, 2013.

[12] M. Farina, P. Colaneri, and R. Scattolini. Block-wise discretization accounting for structural constraints. *Automatica*, 49(11):3411–3417, 2013.

[13] M. Farina and R. Scattolini. An output feedback distributed predictive control algorithm. In *Proc. IEEE Conference on Decision and Control*, pages 8139–8144, 2011.

[14] M. Farina and R. Scattolini. An output feedback distributed predictive control algorithm. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 8139 –8144, 2011.

[15] M. Farina and R. Scattolini. Distributed predictive control: a non-cooperative algorithm with neighbor-to-neighbor communication for linear systems. *Automatica*, 48(6):1088–1096, 2012.

[16] K.H. Johansson and J.L.R. Nunes. A multivariable laboratory process with an adjustable zero. In *Proc. American Control Conference*, volume 4, pages 2045–2049. IEEE, 1998.

[17] J. Lavaei, A. Momeni, and A. G. Aghdam. A model predictive decentralized control scheme with reduced communication requirement for spacecraft formation. *IEEE Transactions on Control Systems Technology*, 16(2):268–278, 2008.

[18] J. Lunze. *Feedback Control of Large Scale Systems*. Prentice Hall, 1992.

[19] D.Q. Mayne, M.M. Seron, and S. V. Rakovic. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41:219–224, 2005.

[20] M. Mercangöz and F.J. Doyle III. Distributed model predictive control of an experimental four-tank system. *Journal of Process Control*, 17(3):297–308, 2007.

[21] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli. The e-puck, a robot designed for education in engineering. In *Proc. Conference on Autonomous Robot Systems and Competitions*, volume 1, pages 59–65, 2009.

[22] R. R. Negenborn and J. M. Maestre eds. *Distributed model predictive control made easy*, volume 310. Springer, 2014.

[23] G. Oriolo, A. De Luca, and M. Vendittelli. WMR control via dynamic feedback linearization: design, implementation, and experimental validation. *IEEE Trans. on Control Systems Technology*, 10(6):835 – 852, nov 2002.

[24] J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, Wisconsin, 2009.

[25] R. Scattolini. Architectures for distributed and hierarchical model predictive control–a review. *Journal of Process Control*, 19(5):723–731, 2009.

[26] Y. Shoham and K. Leyton-Brown. *Multiagent systems: algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2009.

[27] D.D. Siljak. *Decentralized Control of Complex Systems*. Academic Press, Cambridge, 1991.

[28] B. Stenlund and A. Medvedev. Level control of cascade coupled flotation tanks. *Control Engineering Practice*, 10(4):443–448, 2002.