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

Marcello Farina *, Giulio Betti, Luca Giulioni, Riccardo Scattolini
Dipartimento di Elettronica, Informazione e Bioingegneria, 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 \( \mathbf{v} = (v_1, \ldots, v_s) \) denotes a column vector with \( s \) (not necessarily
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 \(s_i\) and \(a, b \in \mathbb{N}, a < b\), we denote \((s_{a+1}, s_{a+2}, \ldots, s_b)\) 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 linear, discrete-time, non-overlapping sub-systems, dynamically coupled through states and inputs, and subject to state and control constraints. For each subsystem \(\mathcal{S}_i\), the dynamics is given by

\[
x_{k+1}^{[i]} = A_{i} x_{k}^{[i]} + B_{i} u_{k}^{[i]} + \sum_{j=1, j \neq i}^{M} \{A_{ij} x_{k}^{[j]} + B_{ij} u_{k}^{[j]}\} + d_{k}^{[i]} \quad (1)
\]

where \(x_{k}^{[i]} \in \mathcal{S}_i \subseteq \mathbb{R}^{m_i}\) and \(u_{k}^{[i]} \in \mathcal{U}_i \subseteq \mathbb{R}^{n_i}\) are the state and input vectors of the \(i\)-th subsystem \(\mathcal{S}_i\) \((i = 1, \ldots, M)\), \(d_{k}^{[i]} \in \mathcal{D}_i \subseteq \mathbb{R}^{m_i}\) is an unknown bounded disturbance and the sets \(\mathcal{S}_i, \mathcal{U}_i\) and \(\mathcal{D}_i\) are convex neighborhoods of the origin. The subsystem \(\mathcal{S}_i\) is said to be a neighbor of the subsystem \(\mathcal{S}_j\) if and only if \(A_{ij} \neq 0\) and/or \(B_{ij} \neq 0\), i.e., if and only if the states \(x_j\) and/or inputs \(u_j\) of \(\mathcal{S}_j\) influence the dynamics of \(\mathcal{S}_i\). The symbol \(\mathcal{N}\) denotes the set of neighbors of \(\mathcal{S}_i\) (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 \(x_k = (x_{k}^{[1]}, \ldots, x_{k}^{[M]}), u_k = (u_{k}^{[1]}, \ldots, u_{k}^{[M]}\) and \(d_k = (d_{k}^{[1]}, \ldots, d_{k}^{[M]}\) the overall collective system can be written as

\[
x_{k+1} = A x_k + B u_k + d_k \quad (2)
\]

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

Assumption 1 There exists a block diagonal matrix \(K = \text{diag}(K_1, \ldots, K_M)\), with \(K_i \in \mathbb{R}^{m_i \times m_i}\), \(i = 1, \ldots, M\) such that: (i) \(A + BK\) is Schur; (ii) \(F_{ii} = (A_i + B_i K_i)\) is Schur, \(i = 1, \ldots, M\).

3 Description of the approach

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

\[
x_{k+1}^{[i]} = A_{i} x_{k}^{[i]} + B_{i} u_{k}^{[i]} + \sum_{j \in \mathcal{N}_i} \{A_{ij} \hat{x}_{k+1}^{[j]} + B_{ij} \hat{u}_{k}^{[j]}\} + w_{k}^{[i]} \quad (3)
\]

where

\[
w_{k}^{[i]} = \sum_{j \in \mathcal{N}_i} \{A_{ij} (x_k^{[j]} - \hat{x}_k^{[j]} + B_{ij} u_k^{[j]} - \hat{u}_k^{[j]}\}) + d_{k}^{[i]} \in \mathcal{W}_i
\]

and \(\mathcal{W}_i = \bigoplus_{j \in \mathcal{N}_i} \{A_{ij} \delta_{j} + B_{ij} \delta_{j}^{[u]}\} \oplus \mathcal{D}_i\).

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 \(\sum_{j \in \mathcal{N}_i} \{A_{ij} \hat{x}_{k+1}^{[j]} + B_{ij} \hat{u}_{k}^{[j]}\}\) can be interpreted as an input known in advance over the prediction horizon \(v = 0, \ldots, N - 1\) to be suitably compensated and \(w_{k}^{[i]}\) is a bounded disturbance to be rejected. By definition, \(w_{k}^{[i]}\) 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 maximin 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 \(\mathcal{S}_i\) is associated to equation (3)

\[
\hat{x}_{k+1}^{[i]} = A_{i} x_{k}^{[i]} + B_{i} u_{k}^{[i]} + \sum_{j \in \mathcal{N}_i} \{A_{ij} \hat{x}_{k}^{[j]} + B_{ij} \hat{u}_{k}^{[j]}\} \quad (4)
\]

while the control law to be used for \(\mathcal{J}_i\) is

\[
u_{k}^{[i]} = \hat{u}_{k}^{[i]} + K_i (x_k^{[i]} - \hat{x}_k^{[i]}\) \quad (5)
\]

where \(K_i\) must be chosen to satisfy Assumption 1.
Letting \( \mathbf{z}_k^{[i]} = \mathbf{x}_k^{[i]} - \bar{\mathbf{x}}_k^{[i]} \), in view of (3), (4), and (5) one has

\[
x_k^{[i]} = F_i \mathbf{z}_k^{[i]} + \mathbf{w}_k^{[i]}
\]

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

### 3.1 The online phase: the i-DPC optimization problems

At any time instant \( k \) each subsystem \( \mathcal{J}_i \) solves the following i-DPC problem.

\[
\begin{align*}
\min_{\mathbf{x}_k^{[i]} \in \mathcal{X}_i, \mathbf{u}_k^{[i]} \in \mathcal{U}_i, \mathbf{z}_k^{[i]} \in \mathcal{Z}_i} & \sum_{v=0}^{N-1} \left( \| \mathbf{x}_{k+v}^{[i]} \|_{Q_i}^2 + \| \mathbf{u}_{k+v}^{[i]} \|_{R_i}^2 + \| \mathbf{z}_{k+v}^{[i]} \|_{P_i}^2 \right) \\
\text{subject to} & (4), \\
& \mathbf{x}_k^{[i]} - \bar{\mathbf{x}}_k^{[i]} \in \mathcal{Z}_i
\end{align*}
\]

(7) subject to (4),

\[
\mathbf{x}_k^{[i]} - \bar{\mathbf{x}}_k^{[i]} \in \mathcal{Z}_i
\]

and, for \( v = 0, \ldots, N-1 \)

\[
\begin{align*}
\mathbf{x}_{k+v}^{[i]} - \mathbf{x}_{k+v}^{[i]} & \in \mathcal{E}_i \\
\mathbf{u}_{k+v}^{[i]} - \mathbf{u}_{k+v}^{[i]} & \in \mathcal{U}_i \\
\mathbf{z}_{k+v}^{[i]} & \in \mathcal{X}_i \subseteq \mathcal{X}_i \mathcal{Z}_i, \\
\mathbf{u}_{k+v}^{[i]} & \in \mathcal{U}_i \subseteq \mathcal{U}_i \mathcal{K}_i \mathcal{Z}_i
\end{align*}
\]

and to the terminal constraint

\[
\mathbf{x}_{k+N}^{[i]} \in \mathcal{X}_i^F
\]

(13)

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

**Assumption 2** Letting \( \mathcal{X}_i = \prod_{i=1}^M \mathcal{X}_i, \mathcal{U}_i = \prod_{i=1}^M \mathcal{U}_i \), and \( \mathcal{X}_i^F = \prod_{i=1}^M \mathcal{X}_i^F \), it holds that:

1. \( \mathcal{X}_i^F \subseteq \mathcal{X}_i \) is an invariant set for \( \bar{\mathbf{x}}_{k+1}^{[i]} = (A + BK) \bar{\mathbf{x}}_k^{[i]}; \)
2. \( \mathbf{u} = K \bar{\mathbf{x}} \in \mathcal{U}_i \) for any \( \bar{\mathbf{x}} \in \mathcal{X}_i^F; \)
3. for all \( \bar{\mathbf{x}} \in \mathcal{X}_i^F \), and for a given constant \( \kappa > 0, \)

\[
\begin{align*}
\mathbf{V}^F (\bar{\mathbf{x}}_{k+1}) - \mathbf{V}^F (\bar{\mathbf{x}}_k) & \leq - (1 + \kappa) \ell (\bar{\mathbf{x}}_k, K \bar{\mathbf{x}}_k) \\
\end{align*}
\]

where \( \mathbf{V}^F (\bar{\mathbf{x}}) = \sum_{i=1}^M \mathbf{V}_i^F (\bar{\mathbf{x}}_i^o) = \sum_{i=1}^M \| \bar{\mathbf{x}}_i^o \|^2_{Q_i^o} + \| \bar{\mathbf{u}}_i^o \|^2_{R_i^o} \) and \( \ell (\bar{\mathbf{x}}, \bar{\mathbf{u}}) = \sum_{i=1}^M \ell_i (\bar{\mathbf{x}}_i^o, \bar{\mathbf{u}}_i^o) = \sum_{i=1}^M (\| \bar{\mathbf{x}}_i^o \|^2_{Q_i^o} + \| \bar{\mathbf{u}}_i^o \|^2_{R_i^o}). \)

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

\[
\mathbf{u}_k^{[i]} = \mathbf{u}_k^{[i]} + K_i (\mathbf{x}_k^{[i]} - \mathbf{z}_k^{[i]})
\]

(15)

Denoting by \( \mathbf{z}_k^{[i]} \) the state trajectory of system (4) stemming from \( \mathbf{z}_k^{[i]} \) and \( \mathbf{u}_k^{[i]} \), at time \( k \) it is also possible to compute \( \mathbf{x}_k^{[i]} \) and \( K_i \mathbf{x}_k^{[i]} \). 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

\[
\bar{\mathbf{x}}_{k+N}^{[i]} = \mathbf{x}_k^{[i]} - \mathbf{z}_k^{[i]} \subseteq \mathcal{X}_i^N + \mathbf{K}_i \bar{\mathbf{x}}_k^{[i]}
\]

(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 \( \mathcal{J}_i \) computes \( \bar{\mathbf{x}}_k^{[i]} \) and \( \mathbf{u}_k^{[i]} \) according to (16) and transmits their values to all the subsystems having \( \mathcal{J}_i \) 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 \( K_i \) satisfying Assumption 1, the sets \( \mathcal{E}_i, \Delta \mathcal{E}_i, \mathcal{U}_i, \mathcal{U}_i \), and the weighting matrices \( Q_i^o, R_i^o, P_i^o \), 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, altering 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.
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 problem and the tuning parameters 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

$$ x_{k+1}^{[i]} = A_{ii} x_k^{[i]} + B_{ii} u_k^{[i]} + \sum_{j=1, j \neq i}^{M} \{ A_{ij} x_k^{[j]} + B_{ij} u_k^{[j]} \} $$

$$ y_k^{[i]} = C_i x_k^{[i]} $$

(17)

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

$$ x_{k+1}^{[i]} = A_{ii} x_k^{[i]} + B_{ii} u_k^{[i]} + \sum_{j=1, j \neq i}^{M} \{ A_{ij} x_k^{[j]} + B_{ij} u_k^{[j]} \} - L_i (y_k^{[i]} - C_i x_k^{[i]}) $$

(18)

Assume that the decentralized observer is convergent i.e., $A + LC$ is Schur, where $C = \text{diag}(C_1, \ldots, C_M)$ and $L = \text{diag}(L_1, \ldots, L_M)$. Under this assumption it is possible to guarantee that the estimation error for each subsystem is bounded, i.e., $x_k^{[i]} - x_t^{[i]} \in \Sigma$, for all $i = 1, \ldots, M$. In this way (18) exactly corresponds with the perturbed system (1), where $d_k^{[i]} = -L_i (y_k^{[i]} - C_i x_k^{[i]})$ is regarded as a bounded disturbance, i.e., $d_k^{[i]} \in \mathcal{G}_i = -L_i C_i \Sigma$. 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 $\Sigma_i$ in such a way that $\Sigma = \bigcap_{i=1}^{M} \Sigma_i$ 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 $y_t^{[i]}$, for all $i = 1, \ldots, M$. Under standard assumptions on the system matrices $(A, B, C)$ (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 $x$ and input $u$ can be computed as follows

$$ \begin{bmatrix} x \\ u \end{bmatrix} = \begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ y_t \end{bmatrix} $$

(19)

where $y_t = (y_t^{[1]} \ldots y_t^{[M]})$. In this way, the setpoint values $\bar{x}^{[i]}$ and $\bar{u}^{[i]}$ for $x_t^{[i]}$ and $u_t^{[i]}$, respectively, are obtained as the vector components of $\bar{x}$ and $\bar{u}$ 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 \((\delta x^i, e^i)\) and whose input variable is \(\delta u^i\), where \(\delta x^i = x^i_k - x^i_{k-1}\), \(e^i = y^i_k - \hat{y}^i\), and \(\delta u^i = u^i_k - u^i_{k-1}\). 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 \(\hat{y}^i\) and \(\hat{u}^i\). 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 \(x^i\) and on \(u^i\) translate into constraints on \(\delta x^i\), \(e^i\), and \(\delta u^i\), 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 \(\hat{y}^i\) given the “ideal” set-points \(\hat{y}^i\), while a reference state and input trajectory layer determines the corresponding state and control trajectories \(\hat{x}^i\) and \(\hat{u}^i\). 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 \(x^i_k\) and \(u^i_k\) of the subsystems as close as possible to \(\hat{x}^i_k\), \(\hat{u}^i_k\), 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 \(\hat{y}^i\) 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.

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 room 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 \(k' = 1 \text{ Wm}^{-2}\text{K}^{-1}\), the one between rooms \(A\) - \(B\) and \(C\) - \(D\) is \(k' = 2.5 \text{ Wm}^{-2}\text{K}^{-1}\), and the one between each room and the external environment is \(k'' = 0.5 \text{ Wm}^{-2}\text{K}^{-1}\).

The nominal external temperature is \(T_e = 0^\circ\text{C}\) and, for the sake of simplicity, solar radiation is not considered. The volume of each room is \(V = 48 \text{ m}^3\), and the wall surfaces between the rooms are all equal to \(s = 12 \text{ m}^2\), while those of the external walls are equal to \(s_e = 24 \text{ m}^2\). Air density and heat capacity are \(\rho = 1.225 \text{ Kg/m}^3\) and \(c = 1005 \text{ J/KgK}\), respectively.

Letting \(\phi = \rho c^V\), the dynamic model is the following:

\[\phi \frac{dT_a}{dt} = s_A k_A^1 (T_b - T_A) + s_A k_A^2 (T_c - T_A) + s_A k_A^3 (T_e - T_A) + q_A\]
\[\phi \frac{dT_b}{dt} = s_B k_B^1 (T_A - T_B) + s_B k_B^2 (T_D - T_B) + s_B k_B^3 (T_e - T_B) + q_B\]
\[\phi \frac{dT_c}{dt} = s_C k_C^1 (T_A - T_C) + s_C k_C^2 (T_D - T_C) + s_C k_C^3 (T_e - T_C) + q_C\]
\[\phi \frac{dT_d}{dt} = s_D k_D^1 (T_B - T_D) + s_D k_D^2 (T_C - T_D) + s_D k_D^3 (T_e - T_D) + q_D\]

The considered equilibrium point is: \(q_A = q_B = q_C = q_D = \dot{q} = \bar{T} s_A k_A^1\), with \(T_A = T_B = T_C = T_D = \bar{T} = 20^\circ\text{C}\). Let \(\Delta T_A = T_A - \bar{T}\), \(\Delta T_B = T_B - \bar{T}\), \(\Delta T_c = T_C - \bar{T}\), \(\Delta T_D = T_D - \bar{T}\), \(\Delta T_e = T_e - \bar{T}\), \(\delta q_A = (q_A - \bar{q})/\rho c^V\), \(\delta q_B = (q_B - \bar{q})/\rho c^V\), \(\delta q_C = (q_C - \bar{q})/\rho c^V\) and \(\delta q_D = (q_D - \bar{q})/\rho c^V\). In this way, denoting \(\sigma_1 = s_A k_A^1/\rho c^V\), \(\sigma_2 = s_B k_B^1/\rho c^V\), \(\sigma_3 = s_C k_C^1/\rho c^V\), \(\sigma_4 = s_D k_D^1/\rho c^V\), \(\sigma_5 = \Delta T_A/\rho c^V\), \(\sigma_6 = \Delta T_B/\rho c^V\), \(\sigma_7 = \Delta T_C/\rho c^V\), \(\sigma_8 = \Delta T_D/\rho c^V\), \(\mu = (\delta q_A, \delta q_B, \delta q_C, \delta q_D)\) and \(d = [\sigma_1, \sigma_2, \sigma_3, \sigma_4]^T\Delta T_e\) the previous model is rewritten in state space representation.

![Fig. 1. Distributed architecture for tracking reference signals.](image-url)
In the simulation, 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:

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

where

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

The discrete-time system of the form (2) (with $\sigma$ and $\sigma_1$) can be obtained by discretization [12] with sampling time $h = 10$. The partition of inputs and states is:

$$x^{[1]} = [\delta T_A \delta T_B]^T, \quad u^{[1]} = [\delta q_A \delta q_B]^T$$

$$x^{[2]} = [\delta T_C \delta T_D]^T, \quad u^{[2]} = [\delta q_C \delta q_D]^T$$

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

$$x^{[1]}_{\min} = [-5 -5]^T, \quad x^{[1]}_{\max} = [5 5]^T$$

$$x^{[2]}_{\min} = [-5 -5]^T, \quad x^{[2]}_{\max} = [5 5]^T$$

$$u^{[1]}_{\min} = [-0.038 -0.038]^T, \quad u^{[1]}_{\max} = [0.030 0.030]^T$$

$$u^{[2]}_{\min} = [-0.038 -0.038]^T, \quad u^{[2]}_{\max} = [0.030 0.030]^T$$

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 $t = 350 s$, 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. 2. Schematic representation of a building with two apartments.](image)

![Fig. 3. State trajectories with DPC (black lines) and cMPC (gray lines) of $\delta T_A$ (left, solid lines), $\delta T_B$ (left, dashed lines), $\delta T_C$ (right, solid lines), $\delta T_D$ (right, dashed lines).](image)

![Fig. 4. Input trajectories with DPC (black lines) and cMPC (gray lines) of $\delta q_A$ (left, solid lines), $\delta q_B$ (left, dashed lines), $\delta q_C$ (right, solid lines), $\delta q_D$ (right, dashed lines).](image)

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

$$ISRE = \sum_{i=1}^{M} \int_{0}^{T_{end}} \sqrt{\dot{x}(t) x(t)} \, dt$$

$$J = \sum_{i=1}^{M} \sum_{k=0}^{N_{end}} x_k^{[i]} T R x_k^{[i]} + u_k^{[i]} Q u_k^{[i]}$$

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].
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 valves \( v_1 \) 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 \( \gamma_1 \) 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

\[
\begin{align*}
\frac{dh_1}{dt} &= -a_1 \sqrt{2gh_1} + \frac{a_4}{A_1} \sqrt{2gh_4} + \gamma_1 A_1 v_1, \\
\frac{dh_2}{dt} &= -a_1 \sqrt{2gh_2} - \frac{a_2}{A_2} \sqrt{2gh_4} - \gamma_1 A_1 v_1, \\
\frac{dh_3}{dt} &= -a_1 \sqrt{2gh_3} + \frac{a_2}{A_2} \sqrt{2gh_4} + \gamma_2 A_2 v_2, \\
\frac{dh_4}{dt} &= -a_1 \sqrt{2gh_4} + \gamma_2 A_2 v_2,
\end{align*}
\]

(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: \( A_1 = A_4 = 28 \text{ cm}^2 \), \( A_2 = A_3 = 32 \text{ cm}^2 \), \( a_1 = a_4 = 0.071 \text{ cm}^2 \), \( a_2 = a_3 = 0.057 \text{ cm}^2 \), \( k_1 = 3.35 \text{ cm}^2/\text{V}s \), \( k_2 = 3.33 \text{ cm}^2/\text{V}s \), \( \gamma_1 = 0.7 \), \( \gamma_2 = 0.6 \). The considered equilibrium point is \( \bar{v}_1 = \bar{v}_2 = 3 \text{ V}, \bar{h}_1 = 12.263 \text{ cm}, \bar{h}_2 = 1.409 \text{ cm}, \bar{h}_3 = 12.783 \text{ cm}, \) and \( \bar{h}_4 = 1.634 \text{ cm} \). Denoting \( \delta h_i = h_i - \bar{h}_i \), \( i = 1, 2, 3, 4 \) 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 \( h = 1 \text{ s} \), we obtain a linear system of the type (2), where

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

The inputs and states are partitioned as:

\[
\mathbf{x}^{[1]} = \begin{bmatrix} \delta h_1 \\ \delta h_2 \end{bmatrix}, \quad \mathbf{u}^{[1]} = \delta v_1
\]

\[
\mathbf{x}^{[2]} = \begin{bmatrix} \delta h_3 \\ \delta h_4 \end{bmatrix}, \quad \mathbf{u}^{[2]} = \delta v_2
\]

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

\[
\mathbf{x}^{[1]}_{\min} = [-12.263 -1.409]^T, \quad \mathbf{x}^{[1]}_{\max} = [40 \ 40]^T + \mathbf{x}^{[1]}_{\max}
\]

\[
\mathbf{x}^{[2]}_{\min} = [-12.783 -1.634]^T, \quad \mathbf{x}^{[2]}_{\max} = [40 \ 40]^T + \mathbf{x}^{[2]}_{\max}
\]

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

\[
\mathbf{x}(0) = \begin{bmatrix} \delta h_1 \\ \delta h_2 \\ \delta h_3 \\ \delta h_4 \\ \delta v_1 \\ \delta v_2 \end{bmatrix} = \begin{bmatrix} 0.274 \ 0.067 \ 0.203 \ 0.254 \ 0 \ 0 \end{bmatrix}
\]

\[
\delta h_i \text{ cm}, \quad \delta v_i \text{ cm}, \quad i = 1, 2, 3, 4
\]

\[
\text{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 \text{ 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 \text{ and } J \text{ defined in (20), (21) and reported in Table 5.2.}
\]

---

<table>
<thead>
<tr>
<th>ISRE</th>
<th>cMPC</th>
<th>461.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPC</td>
<td>501.1</td>
<td></td>
</tr>
<tr>
<td>DPC/cMPC</td>
<td>1.09</td>
<td></td>
</tr>
</tbody>
</table>

Table 1

ISRE and J with DPC and cMPC in the temperature control problem.
the manipulated inputs are the commands to the flotation tanks proposed in [28]. The system is constituted by the tanks. The manipulated inputs are the commands to the flotation tanks proposed in [28]. The system is constituted by the tanks.

The mathematical model describing the four tanks. Note that, also in this case, the distributed control point and its discretization with mE-ZOH using a sampling time 5 s, leads to a linear system of the form (2), where

\[ B_d = [1.4714 0 0 0 0]^T \]

and

\[ A = \begin{bmatrix} 0.853 & 0.147 & 0 & 0 & 0 \\ 0.136 & 0.727 & 0.136 & 0 & 0 \\ 0 & 0.136 & 0.727 & 0.136 & 0 \\ 0 & 0 & 0 & 0.157 & 0.969 \\ -0.104 & 0 & 0 & 0 & 0 \\ 0.096 & -0.096 & 0 & 0 & 0 \\ 0 & 0 & 0.096 & -0.096 & 0 \\ 0 & 0 & 0 & 0.111 & -0.248 \end{bmatrix} \]

The partitions of inputs and states, for \( i = 1, \ldots, 5 \) is:

\[ x_i = \delta y_i, u_i = \delta v_i \]

The constraints on the inputs and the states of the linearized system, for \( i = 1, \ldots, 5 \), 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 \( \delta y_1 = -23.3 \text{ cm} \), \( \delta y_2 = -21.6 \text{ cm} \), \( \delta y_3 = 23.3 \text{ cm} \), \( \delta y_4 = 44.4 \text{ cm} \), and \( \delta y_5 = -12.9 \text{ cm} \) at time \( t = 300 \text{s} \). A disturbance of magnitude \( w = 0.1 \text{ m}^3/\text{s} \) 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.
The dynamics of a single robot is described by a modified version of the first-order kinematic model [23]:

\[
\begin{align*}
\dot{x} &= v \cos \phi \\
\dot{y} &= v \sin \phi \\
\dot{\phi} &= \omega \\
\dot{v} &= a
\end{align*}
\]  

(24a-d)

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 measured variables are \(x, y, \) and \( \eta_1 = x, \eta_2 = \dot{x}, \eta_3 = y, \eta_4 = \dot{y} \), and the dynamics resulting from (24) is

\[
\begin{align*}
\dot{\eta}_1 &= \eta_2 \\
\dot{\eta}_2 &= a \cos \phi - v \omega \sin \phi \\
\dot{\eta}_3 &= \eta_4 \\
\dot{\eta}_4 &= a \sin \phi + v \omega \cos \phi
\end{align*}
\]  

(25a-d)

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 \((a_x, a_y)\) from \((\omega, a)\) compute

\[
\begin{bmatrix}
\omega \\
\alpha
\end{bmatrix} = \frac{1}{v} \begin{bmatrix}
-\sin \phi & \cos \phi \\
\cos \phi & \sin \phi
\end{bmatrix} \begin{bmatrix}
a_x \\
a_y
\end{bmatrix}
\]  

(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

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

The measured variables are \(x\) and \(y\), i.e., \(\eta_1\) and \(\eta_3\) 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 cm² working area. Position and orientation of each robot are detected using two colored circles, placed on the top of each agent, see Figure 12.

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.

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


