

Distributed MPC for Large-Scale Systems



Marcello Farina and Riccardo Scattolini

1 Introduction and Motivations

In the past decades a number of popular methods have been developed for control of a multi-input and multi-output system \mathcal{S}^o , including optimal LQ control theory, pole-placement methods, $\mathcal{H}_2/\mathcal{H}_\infty$ control, and Model Predictive Control (MPC). These methods are intrinsically of centralized nature, i.e. the vector of control actions \mathbf{u} is computed based on the knowledge of the whole state \mathbf{x} or output \mathbf{y} vectors. This allows to guarantee properties, in terms of stability, robustness, and performance. However, these methods display many limitations in case of large-scale [1] or complex [2] systems, i.e., systems with a large size, often characterized by the cooperation of many different parts (e.g., machines, reactors, robots, transportation systems), and possibly by uncertainties on the system components. Just to mention the main issues:

- the actuators and the transducers may be highly geographically distributed, and this may bring about *transmission* delays or failure issues.
- The control problem grows in size with the dimensionality of the system, and the related computational burden may, in turn, grow significantly. This is particularly true for methods, like MPC, where an optimization problem must be solved at any new sampling time. In turn, these scalability issues may induce large - and even inadmissible - computational delays, with serious limitations on the size of practically tractable problems.
- Single components or subsystems can be subject to structural changes, failures, and some may be removed, added, or replaced. Centralized control systems are normally non-robust and non-flexible with respect to such occurrences. This issue may have a big economic impact, since a new design and implementa-

M. Farina · R. Scattolini (✉)

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano,
via Ponzio 34/5, 20133 Milano, Italy
e-mail: marcello.farina@polimi.it; riccardo.scattolini@polimi.it

tion phase for the overall control system must be carried out, with significant expenses, e.g., due to the required downtime.

All the above reasons motivate the development of new methods for the design of more flexible control structures where the centralized controller is replaced by a set of M local regulators, possibly coordinated to recover to the maximum extent the performance guaranteed by a centralized solution. According to a terminology nowadays well accepted, we will define *decentralized control structures* those where the local regulators are fed by independent subsets of states and/or outputs and do not communicate with each other. As a middle-ground solution between centralized and decentralized control, we will denote by *distributed control structures* the schemes where the M local regulators are fed by not necessarily independent subsets of states and/or outputs and can exchange information to coordinate their actions. A pictorial representation of centralized, decentralized, and distributed control structures in the case $M = 2$ is reported in Figure 1.

The possibility to coordinate and negotiate the local control actions provided by distributed schemes and made possible thanks to suitable transmission networks fits very well with an optimization-based framework, where many ideas of game theory can be applied. Therefore, MPC is probably the best advanced control approach for the synthesis of distributed control algorithms and for this reason the aim of this chapter is to present in plain form the main ideas underlying some of the most popular Distributed MPC (DMPC) algorithms. Since nowadays many survey papers and books are dedicated to DMPC, this chapter is not aimed to provide an extensive review of all the available DMPC algorithms, for which the reader is referred to [3, 4], but rather to highlight the main features, properties, and requirements of the major classes of DMPC methods. With the goal to simplify the presentation some restrictive choices will be made: (i) the theoretical properties of the considered methods will not be examined in detail, and the reader will be referred to the relevant liter-

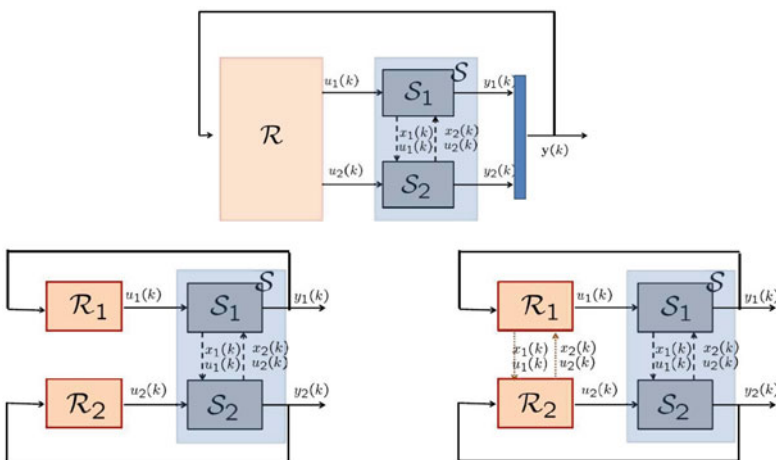


Fig. 1: Centralized (top panel), decentralized (bottom left panel), and distributed (bottom right panel) control structures.

ature; (ii) purely regulation problems will be considered; (iii) the system \mathcal{S}^o to be controlled will be assumed to be described by a linear time-invariant discrete-time model. Some of these assumptions will be re-examined in the final section of the chapter.

2 Model and Control Problem Decomposition

As it is claimed in [1], “if such (large-scale) systems must be controlled, (...) they necessitate new ideas for decomposing and dividing the analysis and control problems of the overall system into rather independent subproblems.” Under this viewpoint the decomposition of the dynamic large-scale system model and of the control problem is a preliminary - but key - step for the development of well-posed decentralized and distributed control procedures. In particular, it is definitely worth noting that the adopted model decomposition has a strong impact on the features, properties, and requirements of the control scheme which is designed based on it.

2.1 Model Decomposition

We assume that the large-scale system \mathcal{S}^o is described by the following linear, discrete-time model

$$\begin{aligned} \mathbf{x}^o(k+1) &= \mathbf{A}^o \mathbf{x}^o(k) + \mathbf{B}^o \mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}^o \mathbf{x}^o(k) \end{aligned} \quad (1)$$

where $\mathbf{x}^o \in R^n$, $\mathbf{u} \in R^m$, $\mathbf{y} \in R^p$ and, unless otherwise specified, the state \mathbf{x}^o will be assumed to be measurable. The state, output, and/or input variables must satisfy constraints of the general type

$$\mathbf{x}^o \in \mathcal{X}^o \quad , \quad \mathbf{y} \in \mathcal{Y} \quad , \quad \mathbf{u} \in \mathcal{U} \quad (2)$$

where \mathcal{X}^o , \mathcal{Y} , and \mathcal{U} are closed sets of proper dimensions containing the origin.

In order to design M decentralized or distributed MPC regulators guaranteeing the stability of the origin of the corresponding closed-loop system, the centralized model (1) must be decomposed into M small scale models \mathcal{S}_i , also denoted model partitions. The first problem is that of partitioning the input and output vectors \mathbf{u} and \mathbf{y} , i.e., to identify, for each subsystem \mathcal{S}_i , a local input vector $u_i \in R^{m_i}$ and a local output vector $y_i \in R^{p_i}$ (commonly, it must hold that $\sum_{i=1}^M m_i = m$ and that $\sum_{i=1}^M p_i = p$). Indeed, a local input/output pair $(u_i(k), y_i(k))$ (also denoted *channel*) must be attributed to each subsystem on the basis of a specific partitioning criterion. Often this is based on physical insight; as an alternative, many methods have been developed and can be adopted to unveil the interactions of MIMO systems. Some of them are based on the analysis of the static and dynamic interactions among inputs and outputs see, e.g., [5, 6].

The state-space model, for each subsystem \mathcal{S}_i , is of the following general type.

$$\begin{aligned} x_i(k+1) &= A_{ii}x_i(k) + B_{ii}u_i(k) + \sum_{j \neq i} (A_{ij}x_j(k) + B_{ij}u_j(k)) \\ y_i(k) &= C_{ii}x_i(k) + \sum_{j \neq i} C_{ij}x_j(k) \end{aligned} \quad (3)$$

where $i = 1, \dots, M$, $x_i \in R^{n_i}$. Many different methods can be used to obtain the models \mathcal{S}_i from \mathcal{S}^o , each corresponding to a different model partition and to a different corresponding interconnection graph. A very rough classification can be made between *non-overlapping* and *overlapping* decompositions: in non-overlapping decompositions the aggregate state of the \mathcal{S}_i model is still of order n , that is the one of the original system \mathcal{S}^o . On the contrary, in overlapping decompositions the overall state of the ensemble of models \mathcal{S}_i is greater than the one of \mathcal{S}^o , i.e. $\sum_{i=1}^M n_i > n$. The characteristics of these two classes will be briefly analyzed in Sections 2.1.1 and 2.1.2.

A critical point regards which constraints should be enforced on the state, output, and/or input variables of \mathcal{S}_i , $i = 1, \dots, M$, to verify the centralized system constraints (2). Note that constraints involving the variables of more than one subsystems (the so-called *coupling* constraints) may be present, and even they may lie at the core of specific types of distributed and coordination control problems. For instance, when subsystems share a common, but limited, input resource some constraints of the type $\sum_{i=1}^M u_i \in \mathcal{U}$ (for a suitably defined \mathcal{U}) may be enforced. On the other hand, a number of power generation devices may be asked to produce a common, but bounded, output, leading to constraints of the type $\sum_{i=1}^M y_i \in \mathcal{Y}$. Another similar example is the case of coordination of moving robots, where collision avoidance constraints are enforced: assuming that the robot - or the robot joints - positions are included in the outputs y_i and that two robots (i.e., \mathcal{S}_1 and \mathcal{S}_2) are involved, these constraints may be formulated as $(y_1, y_2) \in \mathcal{Y}_{12}$, for a suitably-defined set \mathcal{Y}_{12} . In some cases coupling constraints can be verified by enforcing a number of local constraints at the same time, e.g., when $u_i \in \mathcal{U}_i$ for all $i = 1, \dots, M$ involves $\sum_{i=1}^M u_i \in \mathcal{U}$. However, this solution may be overly conservative and highly suboptimal in many contexts and should be discarded, at the price of including coupling (also said *complicating*) constraints in the - distributed - control problem formulation. Summing up, from now on we assume that constraints (2) allow to formulate two types of constraints on the model partition variables: local constraints, to be enforced for all $i = 1, \dots, M$

$$x_i \in \mathcal{X}_i \quad , \quad y_i \in \mathcal{Y}_i \quad , \quad u_i \in \mathcal{U}_i \quad (4)$$

and/or coupling constraints, which will be represented as follows for simplicity

$$(x_1, \dots, x_M) \in \mathcal{X}_C \quad , \quad (y_1, \dots, y_M) \in \mathcal{Y}_C \quad , \quad (u_1, \dots, u_M) \in \mathcal{U}_C \quad (5)$$

Note, in passing, that there may not be a trivial/direct correspondence between the state variable x_i of \mathcal{S}_i , $i = 1, \dots, M$ and the state \mathbf{x}^o of \mathcal{S}^o . Therefore, especially - but not only - when overlapping decompositions are used, it may be critical and/or ambiguous to translate the constraint $\mathbf{x}^o \in \mathcal{X}^o$ into a number constraints on the local state variables x_i .

Subclasses of the representation (3) are called *input-decoupled* when $B_{ij} = 0$ for all i and $j \neq i$ or *state-decoupled* when $A_{ij} = 0$ for all i and $j \neq i$. In a wide class of

control problems, for instance the coordination of independent vehicles with coupling constraints, the subsystems are *dynamically decoupled*, with $A_{ij} = 0$, $B_{ij} = 0$, $C_{ij} = 0$ for all i and $j \neq i$, but coupled through the state or control constraints (5).

Some final comments are in order. First, the model (1) is often computed as the discretization of a continuous time physical system made by the interconnection of subsystems. In this case, the corresponding matrices have a sparse structure which should be maintained after discretization. Unfortunately, if a Zero Order Hold transformation is used, this property is lost. To recover it, one should resort to the forward Euler (fE) discretization approach or to the approximate discretization method described in [7], specifically developed for distributed control.

Secondly, models of type (3) include the information on how subsystems have influence on each other. In other words, if (for $j \neq i$) $A_{ij} \neq 0$ and/or $B_{ij} \neq 0$, the state/input variables of subsystem \mathcal{S}_j directly impact on the dynamics of subsystem \mathcal{S}_i . Consistently, we can define the set of neighbors (or parents) of subsystem \mathcal{S}_i as $\mathcal{N}_i := \{j : \|A_{ij}\| + \|B_{ij}\| \neq 0\}$ and a corresponding direct interconnection graph, which highlights the system-wide dependencies between subsystems. Similar interconnection (possibly undirected) graphs can be drawn when coupling constraints are present, i.e., if a coupling constraint involves a set of subsystems, they should be considered as neighbors.

2.1.1 Non-overlapping Decompositions

Perhaps the most natural choice to decompose \mathcal{S}^o is to partition the state \mathbf{x} into M non-overlapping sub-vectors x_i with $\sum_{i=1}^M n_i = n$, so that, up to a suitable state variable permutation, $\mathbf{x}^o = (x_1, \dots, x_M)$ and that the original system can be written as

$$\begin{bmatrix} x_1(k+1) \\ \vdots \\ x_M(k+1) \end{bmatrix} = \begin{bmatrix} A_{11} & \dots & A_{1M} \\ \vdots & \ddots & \vdots \\ A_{M1} & \dots & A_{MM} \end{bmatrix} \begin{bmatrix} x_1(k) \\ \vdots \\ x_M(k) \end{bmatrix} + \begin{bmatrix} B_{11} & \dots & B_{M1} \\ \vdots & \ddots & \vdots \\ B_{M1} & \dots & B_{MM} \end{bmatrix} \begin{bmatrix} u_1(k) \\ \vdots \\ u_M(k) \end{bmatrix}$$

$$\begin{bmatrix} y_1(k) \\ \vdots \\ y_M(k) \end{bmatrix} = \begin{bmatrix} C_{11} & \dots & C_{1M} \\ \vdots & \ddots & \vdots \\ C_{M1} & \dots & C_{MM} \end{bmatrix} \begin{bmatrix} x_1(k) \\ \vdots \\ x_M(k) \end{bmatrix}$$

It is intuitive that in the definition of the submodels (3) one should partition the original state so that the couplings among the subsystems are reduced as much as possible, i.e. A_{ij} , B_{ij} , C_{ij} , $i \neq j$ should be null or “small” (according to a proper norm or criterion) to the maximum possible extent. In the common practice, the state permutation/partition can be carried out based on the plant physical insight or based on available algebraic algorithms. For example, there exist graph-based methods for reordering the state, input, and output variables in order to highlight inherent structures, e.g., the presence of weakly interacting subsystems (see, e.g., the

ε -nested decomposition proposed in [2]) or cascaded configurations (e.g., the lower-block-triangular LBT decomposition discussed in [2]). Finally, methods have been proposed in the literature [2] for devising suitable changes of coordinates which lead to specific system decompositions (e.g., the *input and/or output decentralized forms*), at the price of a loss of physical insight.

2.1.2 Overlapping Decompositions

In some cases the coupling strength between different subsystems is relevant, which prevents the decomposition into disjoint subsystems to result into an effective control-oriented partition. An alternative to non-overlapping decompositions is to decompose the systems into subsystems which have some equations (and states) in common, i.e., carrying out a so-called *overlapping decomposition*. This may result in obtaining overlapping but weakly coupled subsystems. Overlapping decompositions have been widely studied in the past, mainly in the context of decentralized control, see, e.g., [2].

In the context of DMPC, a number of distributed control methods require the subsystem interconnections to be represented in the following *state-decoupled* form

$$\begin{cases} x_i(k+1) = A_{ii}x_i(k) + \sum_{j=1}^M B_{ij}u_j(k) \\ y_i(k) = C_i x_i(k) \end{cases} \quad (6)$$

However, it is rarely possible to obtain a representation of this type by simply applying a non-overlapping decomposition such as the one described in the previous paragraphs. Therefore, it is possible to adopt a *fully overlapping decomposition* where each subsystem is of full order. The simplest way to obtain this decomposition consists of replicating M times model (1), that is setting, for all $i, j = 1, \dots, M$, $A_{ii} = \mathbf{A}^o$, $B_{ij} = \mathbf{B}^{j,o}$, $C_i = \mathbf{C}^{i,o}$, where $\mathbf{B}^{j,o}$ is the j -th block column of \mathbf{B}^o , while $\mathbf{C}^{j,o}$ is the j -th block row of \mathbf{C}^o ; an alternative procedure is sketched in [8].

In general, overlapping decompositions are non-minimal, since the state dimension does not decrease (for each subsystem) under the application of the proposed partition. However, as better specified in the following, for subsystem \mathcal{S}_i , the local control variable is assumed to be u_i , while the u_j 's, $j \neq i$ are considered as external signals, so that the number of variables to be optimized by the local DMPC algorithm is smaller than in the corresponding centralized problem.

2.2 Partition Properties and Control

The main properties of the model partitions which have a major impact on the resulting decentralized and distributed MPC-based schemes are the following.

- **Minimality of the state representation.** Minimality (in terms of dimension of the state/input/output variables for each subsystem) is required to limit the algo-

rithm computational burden, since the number of involved variables generally directly impacts on the computational demands of the algorithm. Also, minimal representations demand minimal information to be stored by local memory units. This, besides requiring scalable memory load, allows for more flexibility of the resulting control systems. In fact, local model variations at subsystem level may require the re-design of local control systems only.

- **Minimality of the interconnection graph.** As discussed, to reduce the probability of network-induced issues (e.g., transmission delays, channel overloads, and package losses) the communication burden required by the adopted control architecture must be reduced as much as possible. To this end, one should reduce both (i) the number of communication links between subsystems (i.e., aiming to have a sparse supporting information transmission network), and (ii) the amount of communication that they should afford. While (ii) mostly depends on the type of the adopted control scheme (we defer the reader to Section 4 for a discussion on this point), (i) may strongly depend upon the approach taken for model partitioning, since the implementation of distributed schemes commonly requires to be supported by an underlying communication graph consistent with the subsystem interconnection graph.
- **Descriptive capabilities of the models.** Model partitioning may have a negative effect on the descriptive capabilities of the submodels. It would be desirable, in fact, that each local controller has the knowledge on how a control action, taken locally, can impact, not only on the local variables, but also on the variables of the surrounding subsystems. Similarly, a complete information on how the control action taken by other subsystems affects local state variables and outputs may be desired.

It is worth noting that the first and the second requirements are in general conflicting with the third one. Indeed, fully descriptive models, which are the ones that allow for full cooperation between local controllers, are often obtained using overlapping - often fully overlapping - partitions, which are the ones which typically lead to a non-minimal state representation and for which the interconnection graph is maximal.

2.3 MPC Problem Separability

To briefly introduce the optimization problems involved in the decentralized and distributed implementations summarized in this chapter, we first introduce the centralized (non-minimal) model which can be drawn by collecting together all the partitions (3). We define $\mathbf{x} = (x_1, \dots, x_M)$, which corresponds to \mathbf{x}^o , up to a state permutation, only in case of non-overlapping decompositions. Consistently, we can describe the overall large-scale system dynamics with a model of the type

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) \end{aligned} \tag{7}$$

Note that system (7) is nothing else than an *expansion* [2] of the corresponding contracted system (1), since their output free and forced motions must be indeed equal to each other. Starting with standard centralized MPC, the control action at each time instant k is obtained by solving a control problem of type

$$\min_{\mathbf{u}(k), \dots, \mathbf{u}(k+N-1)} J(k) \quad (8a)$$

subject to the transient constraints (7), (4), (5), for times $k, \dots, k+N-1$, and a terminal constraint of the type

$$\mathbf{x}(k+N) \in \mathcal{X}_f \quad (8b)$$

In (8a), $J(k)$ is the cost function, while \mathcal{X}_f in (8b) is the terminal constraint set. As a common ground of the methods discussed in this chapter, problem (8) is required to be separable into a number of subproblems to be solved by local computing units. This first requires that the cost function to be minimized is formally separable, i.e.,

$$J(k) = \sum_{i=1}^M \rho_i J_i(k) \quad (9)$$

where $J_i(k)$ is a positive-definite (quadratic, for simplicity) function of local input and state variables of subsystem \mathcal{S}_i

$$J_i(k) = \sum_{j=0}^{N-1} [\|x_i(k+j)\|_{Q_i}^2 + \|u_i(k+j)\|_{R_i}^2] + \|x_i(k+N)\|_{P_i}^2 \quad (10)$$

and parameter $\rho_i > 0$, where $\sum_{i=1}^M \rho_i = 1$.

Secondly, we need to enforce the terminal constraint (8b) by imposing M local terminal constraints of type $x_i(k+N) \in \mathcal{X}_{f,i}$, and therefore \mathcal{X}_f must be defined as the Cartesian product of M sets $\mathcal{X}_{f,i} \subseteq \mathbb{R}^{n_i}$.

However, separability is not the only requirement for the well-posedness of the problem, but also some assumptions - required for general-type MPC problems [8] - must be fulfilled by the “collective” terminal cost and terminal set, i.e., $V_f = \sum_{i=1}^M \rho_i \|x_i(k+N)\|_{P_i}^2 = \|\mathbf{x}(k+N)\|_P^2$, with $P = \text{diag}(\rho_1 P_1, \dots, \rho_M P_M)$ and \mathcal{X}_f , respectively. More specifically V_f and \mathcal{X}_f must also be a Lyapunov function and a positively invariant set, respectively, for the expanded system (7), controlled using a suitable (and possibly decentralized/distributed) auxiliary control law. The latter requirements are not easily compatible with the separability assumptions: in the sequel, for each of the described methods, we will discuss how they are guaranteed.

3 Decentralized MPC

A simple paradigmatic decentralized method is the one proposed in [9], for large-scale linear processes subject to local input saturations only. The global model (1), with stable matrix \mathbf{A}^o , is approximated by a number of (possibly overlapping) subsystems of type (3) with stable matrices A_{ii} , used for local predictions. The key feature of decentralized control is that there is no communication between the different local controllers, as shown in Figure 1. Therefore a modelling error is introduced by neglecting couplings, i.e., by setting $A_{ij} = 0$ and $B_{ij} = 0$ for all $j \neq i$, $i = 1, \dots, M$. The proposed decentralized MPC algorithm requires that, at each time instant k , the following optimization problem is solved by each computational unit.

$$\min_{u_i(k), \dots, u_i(k+N-1)} J_i(k) \quad (11)$$

subject to the dynamical model

$$x_i(k+1) = A_{ii}x_i(k) + B_{ii}u_i(k) \quad (12)$$

and the local input constraints $u_i(k) \in \mathcal{U}_i$, for times $k, \dots, k+N-1$. Here no terminal constraint is required in view of the fact that no state constraints are imposed and that the local system matrices A_{ii} are stable. Indeed, the auxiliary control law is $u(k) = 0$ and, correspondingly, the separability of the cost function is obtained by selecting P_i in such a way that $A_{ii}^T P_i A_{ii} - P_i = -Q_i$. This, if we neglect the interconnections terms A_{ij} with $j \neq i$, between subsystems, makes V_f (as defined in the previous paragraph) a Lyapunov function for the overall - decentralized - system. The asymptotic stability properties of the control system are proved a posteriori if some inequality conditions are verified. In general, in order to achieve closed-loop stability as well as performance in the development of decentralized MPC algorithms, the interconnections (at least the terms A_{ij}) between different subsystems should be weak or the system should display peculiar structures (e.g., acyclic graphs).

A different approach consists of considering couplings as disturbances. For example, in [10], a decentralized MPC algorithm for nonlinear discrete time systems subject to decaying disturbances was presented. In the design of the decentralized MPC, the effects of interconnections between different subsystems are considered as perturbation terms whose magnitude depends on the norm of the system states. No information is exchanged between the local controllers and the stability of the closed-loop system relies on the inclusion of a contractive constraint in the formulation of each of the decentralized MPC problems. In [11], the stability of a decentralized MPC is analyzed from an input-to-state stability (ISS) point of view. For linear systems this approach consists of rewriting the subsystem \mathcal{S}_i model (3) as

$$x_i(k+1) = A_{ii}x_i(k) + B_{ii}u_i(k) + v_i(k) \quad (13)$$

where $v_i(k) = \sum_{j \neq i} (A_{ij}x_j(k) + B_{ij}u_j(k))$ is regarded as an unknown, but bounded - if local state and input constraints (4) are enforced - disturbance which must be

compensated or rejected using an ad-hoc robust MPC scheme. Along this line, e.g., the algorithm in [12] has been proposed for linear systems, resorting to tube-based MPC (see [13, 14] and Chapter Robust Optimization for MPC). For more details, see also Chapter “Scalable MPC Design”.

It is worth remarking that, in all the decentralized schemes described above, no information is required to be transmitted between local regulators, since only the information regarding the local state is used in the corresponding MPC optimization problem (11).

4 Distributed MPC

According to the taxonomy proposed in [3] and nowadays widely used, DMPC algorithms can be broadly classified as follows:

- *iterative* or *non-iterative*: in iterative algorithms information can be transmitted among the local regulators many times within the sampling time. This opens the way to the design of methods aimed at achieving a global consensus among the regulators on the actions to be taken within the sampling interval. On the contrary, in non-iterative algorithms information is transmitted only once in the sampling period, so that the regulators are required to possess some robustness properties to compensate for the reduced information available.
- *cooperating* and *non-cooperating*: in cooperating algorithms each local regulator tries to minimize a global cost function, so that Pareto - i.e., system-wide - optimal solutions can be computed, at least in principle. In non-cooperating algorithms each regulator minimizes its local cost function, with possible conflicting goals; in this case, Nash equilibria are to be expected.
- *fully connected* or *partially connected*: in fully connected algorithms information is transmitted and received from any local regulator to all the others. In partially connected methods the information is exchanged between any local regulator and a subset of the others. Although this is not a structural property, it can strongly influence the properties and the transmission load of the methods as well as their computational burden.

4.1 Cooperating DMPC

As a prototype algorithm for this class of DMPC methods, we make reference to the results reported in [8, 15]. The system to be controlled is assumed to be in the state decoupled form (6): this implies that the transition matrix of the expanded system (7) is block-diagonal, i.e., $\mathbf{A} = \text{diag}(A_{11}, \dots, A_{MM})$. For simplicity we assume \mathbf{A} to be asymptotically stable. This ensures the separability of the cost function $J(k)$ as in (9): in fact we can take $\mathbf{u}(k) = \mathbf{K}\mathbf{x} = 0$ (i.e., $\mathbf{K} = 0$) as a - decentralized - auxiliary control law for the expanded system and we can select P_i , $i = 1, \dots, M$ in such a

way that $A_i^T P_i A_{ii} - P_i = -Q_i$, $Q_i > 0$. This makes V_f a Lyapunov function for the system $\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k)$, as required. Finally, in the present algorithm, only input constraints are enforced.

At time k all the local MPC control algorithms have knowledge of the overall state $\mathbf{x}(k)$ and of the full system dynamics, meaning that a fully connected communication network is required to support the transmission of the global state to all the local control stations.

The following iterative procedure is performed within the sampling period from time k and time $k+1$:

- at iteration (negotiation) step p , $p \geq 1$, each local controller in \mathcal{S}_i has the information about the possible input sequences of the other subsystems, $u_j^{p-1}(k+l)$, for $l = 1, \dots, N-1$ and $j \neq i$, to be broadcast thanks to the available fully connected communication network; the following (global) optimization problem is solved at a local level

$$\min_{u_i(k), \dots, u_i(k+N-1)} J(k) \quad (14)$$

subject to the expanded model (7) and, for $l = 0, \dots, N-1$,

$$u_i(k+l) \in \mathcal{U}_i \quad (15)$$

$$u_j(k+l) = u_j^{p-1}(k+l), \forall j \neq i \quad (16)$$

- letting $u_i^o(k), \dots, u_i^o(k+N-1)$ be the optimal solution, a convex combination between the solution at the previous iteration and the newly computed one is used, i.e.

$$u_i^p(k+l) = \alpha_i u_i^{p-1}(k+l) + (1 - \alpha_i) u_i^o(k+l), l = 0, \dots, N-1$$

where $\alpha_i \in (0, 1)$ and $\sum_{i=1}^M \alpha_i = 1$.

- if a termination condition, which can depend on the elapsed time within the sampling period or on an optimality test, is satisfied, the iterative procedure ends and the last computed value $u_i^p(k)$ is used as $u_i(k)$, otherwise a new iteration starts ($p \leftarrow p+1$)
- when a new measurement is received, ($k \leftarrow k+1$) the overall procedure is restarted.

The algorithm requires an initialization for $p = 0$, which can be obtained based on the optimal control sequence at the previous time $k-1$.

As shown in [8, 15] stability of the closed-loop system is guaranteed for any number of iterations performed within the sampling time; in addition, it can be proven that the computed solution converges to the one of the corresponding centralized control system as the number of iterations (p) increases. Finally, the method can be extended to cope with unstable open-loop systems, provided that a suitable zero terminal condition is included into the problem formulation, and with tracking problems [8].

Interestingly, in [8] it has been shown that, if each local controller adopts a non-cooperating (selfish) approach and minimizes only its own cost function J_i , convergence to a Nash equilibrium is achieved and no stability guarantees can be proven. In [16], a further step is done: more specifically, it is shown that it is possible to add, for each subsystem, a constraint related to the maximal satisfactory and sufficiently small (denoted *satisficing* in the papers) cost γ_i , i.e., $J_i(k) \leq \gamma_i$. According to [16], this variation allows to shift from a purely cooperating scheme (denoted also *categorical altruist* algorithm) to a scheme (denoted *situational altruist*) where local (*selfish*) constraints are introduced.

4.2 Non-cooperating Robustness-Based DMPC

The algorithm described in [17] is based on the idea that each subsystem i transmits to its neighbors its planned state reference trajectory $\tilde{x}_i(k+j)$, $j = 1, \dots, N$, over the prediction horizon and guarantees that, for all $j \geq 0$, its actual trajectory lies in a “tube” centered in \tilde{x}_i , i.e. $x_i(k+j) \in \tilde{x}_i(k+j) \oplus \mathcal{E}_i$, where \mathcal{E}_i is a compact set including the origin. Then, assuming here for simplicity that the system is input decoupled, Equation (3) can be written as

$$x_i(k+1) = A_{ii}x_i(k) + B_{ii}u_i(k) + \sum_j A_{ij}\tilde{x}_j(k) + w_i(k) \quad (17)$$

where $w_i(k) = \sum_j A_{ij}(x_j(k) - \tilde{x}_j(k)) \in \mathcal{W}_i$ is a bounded disturbance to be rejected using the tube-based MPC approach [13] (see also Chapter “Robust Optimization for MPC”), where $\mathcal{W}_i = \bigoplus_j A_{ij}\mathcal{E}_i$. The term $\sum_j A_{ij}\tilde{x}_j(k)$ is equivalent to a non-manipulable input, known in advance over the prediction horizon, to be properly compensated.

From (17), the i -th subsystem nominal model [13] is defined as

$$\hat{x}_i(k+1) = A_{ii}\hat{x}_i(k) + B_{ii}\hat{u}_i(k) + \sum_j A_{ij}\tilde{x}_j(k) \quad (18)$$

Letting $\mathbf{K} = \text{diag}(K_1, \dots, K_M)$ be a block-diagonal matrix such that both $\mathbf{A} + \mathbf{BK}$ and $A_{ii} + B_{ii}K_i$ are stable, the local control law is chosen as

$$u_i(k) = \hat{u}_i(k) + K_i(x_i(k) - \hat{x}_i(k)) \quad (19)$$

From (17) and (19), letting $z_i(k) = x_i(k) - \hat{x}_i(k)$, it holds that

$$z_i(k+1) = (A_{ii} + B_{ii}K_i)z_i(k) + w_i(k) \quad (20)$$

where $w_i \in \mathcal{W}_i$. Since \mathcal{W}_i is bounded and $A_{ii} + B_{ii}K_i$ is stable, there exists a robust positively invariant set \mathcal{Z}_i for (20) such that, for all $z_i(k) \in \mathcal{Z}_i$ and $w_i(k) \in \mathcal{W}_i$, one has $z_i(k+1) \in \mathcal{Z}_i$. According to the approach developed in [13], given \mathcal{Z}_i and assuming that there exist neighborhoods of the origin $\Delta\mathcal{E}_i$ such that

$$\Delta \mathcal{E}_i \oplus \mathcal{Z}_i \subseteq \mathcal{E}_i \quad (21)$$

at any time instant k the i -th subsystem computes the value of $\hat{u}_i(k)$ in (19) as the solution to

$$\min_{\hat{x}_i(k), \hat{u}_i(k), \dots, \hat{u}_i(k+N-1)} J_i(k) \quad (22a)$$

subject to (18) and the initial state constraint

$$x_i(k) - \hat{x}_i(k) \in \mathcal{Z}_i \quad (22b)$$

For $l = 0, \dots, N-1$, to guarantee that the difference between x_i and \tilde{x}_i is effectively limited as initially stated, we require that

$$\hat{x}_i(k+l) - \tilde{x}_i(k+l) \in \Delta \mathcal{E}_i \quad (22c)$$

Both local (4) and coupling (5) constraints can be imposed. This is done by requiring that, for $l = 0, \dots, N-1$

$$\hat{x}_i(k+l) \in \hat{\mathcal{X}}_i \quad (22d)$$

$$(\tilde{x}_1(k+l), \dots, \hat{x}_i(k+l), \dots, \tilde{x}_M(k+l)) \in \hat{\mathcal{X}}_C \quad (22e)$$

This requires to suitably define the sets $\hat{\mathcal{X}}_i$ and $\hat{\mathcal{X}}_C$ as restricted ones, e.g., by setting $\hat{\mathcal{X}}_i \subseteq \mathcal{X}_i \ominus \mathcal{Z}_i$. Although state constraints only have been defined for simplicity, input constraints can be included similarly. Finally, the scheme calls for the definition of terminal constraints of the type

$$\hat{x}_i(k+N) \in \hat{\mathcal{X}}_{f,i} \quad (22f)$$

With the optimal solution at time k , it is also possible to compute the predicted value $\hat{x}_i(k+N)$, which is used to incrementally define the reference trajectory of the state to be used at the next time instant $k+1$, i.e. $\tilde{x}_i(k+N) = \hat{x}_i(k+N)$.

Condition (21) is a key condition for the well posedness of the present distributed control scheme. Despite its analysis goes beyond the scope of this chapter, it is worth remarking that it is equivalent to the so-called tube-based small gain condition for networks discussed in Chapter “Scalable MPC Design”.

Remarkably, each local control station uses only local state information (i.e., $x_i(k)$) and its neighbors’ planned state trajectories $\tilde{x}_j(k)$. The latter is transmitted in a neighbor-to-neighbor fashion thanks to the available partially connected communication network.

A significant work has been devoted to the proper definition of separable terminal cost and constraint sets. In [17, 18] methods for a proper choice of the weights Q_i , R_i , P_i , of sets \mathcal{E}_i , and of the terminal set $\hat{\mathcal{X}}_{f,i}$ guaranteeing the well posedness and the stabilizing properties of the algorithm are proposed.

4.3 Distributed Control of Independent Systems

A prototype non-iterative algorithm for independent systems coupled through constraints is now described, inspired by the approach described in [19]. The system is assumed to be affected by an unknown, but bounded noise, so that the robust tube-based approach of [13] is used also in this case. The model of the i -th subsystem, $i = 1, \dots, M$, is described by

$$x_i(k+1) = A_{ii}x_i(k) + B_{ii}u_i(k) + d_i(k) \quad (23)$$

where $d_i(k) \in \mathcal{D}_i$ is a bounded disturbance. The M systems are subject to both local and coupling constraints (4) and (5), respectively.

For each system $i = 1, \dots, M$, the local nominal model

$$\hat{x}_i(k+1) = A_{ii}\hat{x}_i(k) + B_{ii}\hat{u}_i(k) \quad (24)$$

is defined and a stabilizing gain K_i is computed. Also in this case, the local stabilizing control law is given by

$$u_i(k) = \hat{u}_i(k) + K_i(x_i(k) - \hat{x}_i(k)) \quad (25)$$

and letting $z_i(k) = x_i(k) - \hat{x}_i(k)$, it holds that

$$z_i(k+1) = (A_{ii} + B_{ii}K_i)z_i(k) + d_i(k) \quad (26)$$

In view of the boundedness of the disturbance and the stability of $A_{ii} + B_{ii}K_i$, there exists a robust positively invariant set \mathcal{Z}_i for (26) such that, for all $z_i(k) \in \mathcal{Z}_i$ and $d_i(k) \in \mathcal{D}_i$, one has $z_i(k+1) \in \mathcal{Z}_i$.

At any time instant k only one system, say the i -th one, is allowed to update its future plans by solving a suitable MPC problem, while all the others update their control variables according to the previously computed control sequence and the corresponding auxiliary law, i.e. their future nominal control moves computed at time k are, for all $j \neq i$

$$\begin{aligned} \hat{u}_j(k+l|k) &= \hat{u}_j(k+l|k-1), \quad l = 0, \dots, N-2 \\ \hat{u}_j(k+N-1|k) &= K_j\hat{x}_j(k+N-1|k-1) \end{aligned}$$

where $\hat{x}_j(k+N-1|k-1)$ is the evolution of the nominal state starting from $\hat{x}_i(k) = x_i(k)$ with the sequence $\hat{u}_j(k+l|k-1)$, $l = 0, \dots, N-2$. We also define $\hat{x}_j(k+N-1|k-1) = (A_{ii} + B_{ii}K_i)\hat{x}_j(k+N-1|k-1)$.

On the contrary, the i -th system computes the value of $\hat{u}_i(k)$ in (19) as the solution to

$$\min_{\hat{x}_i(k), \hat{u}_i(k), \dots, \hat{u}_i(k+N-1)} J_i(k) \quad (27a)$$

subject to (18) and the initial state constraint

$$x_i(k) - \hat{x}_i(k) \in \mathcal{Z}_i \quad (27b)$$

Both local (4) and coupling (5) constraints are forced by requiring that, for $l = 0, \dots, N-1$

$$\hat{x}_i(k+l) \in \hat{\mathcal{X}}_i \quad (27c)$$

$$(\hat{x}_1(k+l|k-1), \dots, \hat{x}_i(k+l) \dots, \hat{x}_M(k+l|k-1)) \in \hat{\mathcal{X}}_C \quad (27d)$$

where the sets $\hat{\mathcal{X}}_i$ and $\hat{\mathcal{X}}_C$ are properly restricted subsets of \mathcal{X}_i and \mathcal{X}_C , e.g., by setting $\hat{\mathcal{X}}_i \subseteq \mathcal{X}_i \ominus \mathcal{Z}_i$. As in the previous algorithms, the scheme calls for the definition of terminal constraints of the type

$$\hat{x}_i(k+N) \in \hat{\mathcal{X}}_{f,i} \quad (27e)$$

Similarly to the non-cooperating robustness-based control scheme presented in Section 4.2, a partially connected communication network is required to support the transmission of the planned trajectories $\hat{x}_j(k+l|k-1)$ to each local control station from the (constraint-based) neighboring ones.

As described in [19, 20], the basic algorithm here described can be greatly enhanced to allow for more than one system updating its future plans with the optimization procedure at each time step. In addition, cooperation is achieved letting each system to minimize a global cost function and the communication requirements can be significantly reduced with respect to an all-to-all solution by exploiting the graph topology forced by the coupling constraints.

Similar schemes have been devised by other research teams, e.g., [21]. The paper [22] also extends this method to cope with economic-based cost functions.

4.4 Distributed Optimization

A different approach with respect to the distributed algorithms previously described consists of computing the optimal solution to the original centralized optimization problem as the iterative solution to smaller, more tractable, and independent ones. This idea, which is the basis of many popular decomposition methods, can be traced back to the early contributions, e.g., [23]. In the context of MPC, the reader is referred to the recent contributions [24–27].

A sketch of a simple version the popular dual decomposition approach, proposed in [24], applied to MPC is now described. Constraints of general type can easily be considered in the present framework, although for simplicity of presentation they will be neglected here. Consider the set of input decoupled systems

$$x_i(k+1) = A_{ii}x_i(k) + B_{ii}u_i(k) + \sum_{j \neq i} A_{ij}x_j(k) \quad (28)$$

and the following centralized problem

$$\min_{u(k), \dots, u(k+N-1)} J(k) \quad (29)$$

In view of the formal separability of the cost function $J(k)$, the coupling between the subproblems is due to the “coupling variables” $v_i = \sum_{j \neq i} A_{ij} x_j$ in (28). Now write Equation (28), similarly to (13), as

$$x_i(k+1) = A_{ii} x_i(k) + B_{ii} u_i(k) + v_i(k) \quad (30)$$

and, denoting by λ_i the Lagrange multipliers, consider the Lagrangian function

$$\mathcal{L}(k) = \sum_{i=1}^M [J_i(k) + \sum_{l=0}^{N-1} \lambda_i(k+l) (v_i(k+l) - \sum_{j \neq i} A_{ij} x_j(k+l))] \quad (31)$$

For the generic vector variable φ , let $\bar{\varphi}_i(k) = [\varphi_i^T(k), \dots, \varphi_i^T(k+N-1)]^T$ and $\bar{\varphi} = [\bar{\varphi}_1^T, \dots, \bar{\varphi}_M^T]^T$. Then, by relaxation of the coupling constraints, the optimization problem of Equation (29) can be stated as

$$\max_{\bar{\lambda}(k)} \min_{\bar{u}(k), \bar{v}(k)} \mathcal{L}(k) \quad (32)$$

or, equivalently

$$\max_{\bar{\lambda}(k)} \sum_{i=1}^M \tilde{J}_i(k) \quad (33)$$

where, letting \bar{A}_{ji} be a block-diagonal matrix made by N blocks, all equal to A_{ji} ,

$$\tilde{J}_i(k) = \min_{\bar{u}_i(k), \bar{v}_i(k)} [J_i(k) + \bar{\lambda}_i^T(k) \bar{v}_i(k) - \sum_{j \neq i} \bar{\lambda}_j^T(k) \bar{A}_{ji} \bar{x}_i(k)] \quad (34)$$

The following two-step iterative procedure is then used at any time step to compute the optimal solution

1. for a fixed $\bar{\lambda}$, solve the set of M independent minimization problems given by Equation (34) with respect to $\bar{u}_i(k), \bar{v}_i(k)$;
2. given the collective values of \bar{u}, \bar{v} computed at the previous step, solve the maximization problem given by (33) with respect to $\bar{\lambda}$. This problem can be solved in a distributed way using a gradient step, see [24].

To summarize, the described iterative algorithm must be supported by a partially connected communication network (for both steps 1 and 2, provided that the latter uses a distributed gradient step). More specifically, at each iteration, for step 1 it is required that, for all $i = 1, \dots, M$, the i -th local computing station receives the current values of $\lambda_j(k+l)$, $l = 1, \dots, N-1$ by the agents $j \neq i$ which have i as neighbor, i.e., such that $i \in \mathcal{N}_j$; on the other hand, for step 2, it is required that the i -th station receives the current values of $x_j(k+l)$ by its neighbors $j \in \mathcal{N}_i$.

It is well recognized that this kind of decomposition approaches are characterized by

slow convergence, due to the great number of iterations required to obtain a solution. However, to this regard, many efficient algorithms have been developed, see, for instance, [26]. In addition, the fundamental properties of recursive feasibility and stability are not a-priori guaranteed, and can be achieved by a proper definition of the optimization problem and of the constraints, see [27].

A final remark is due: as noted above and as apparent from (33), the separability of the cost function $J(k)$ is a major requirement also for this method, as well as - for constrained problems - the separability of the terminal constraint set. The previously discussed approaches can be used to this aim, including the one presented in [17, 18]. Also the recent work [28] is devoted to this problem and allows for scalable design. See Chapter “Scalable MPC Design” for more details.

5 Extensions and Applications

The DMPC algorithms discussed in the previous sections have been designed for linear, discrete-time, and time invariant systems, and the main approaches and ideas behind most of the nowadays available methods for the regulation problem have been described. However, the recent and tumultuous research activity in the field has produced a number of algorithms dealing with a large variety of systems and control problems. Among them, we here recall some of the most interesting research directions, with some references:

- DMPC algorithms have been developed for continuous-time and nonlinear systems in, e.g., [21, 29–31].
- The output feedback case has been studied in [32], while the tracking problem has been analyzed, e.g., in [8, 33]. An alternative approach, based on the particular class of MPC strategies called Command Governor methods, has been reported in [34] and in the papers referenced therein.
- DMPC for systems affected by stochastic noises has been considered in [35–37].
- Economic MPC (see Chapter “Economic Model Predictive Control: Some Design Tools and Analysis Techniques”) has been extended to cope with a distributed implementation in [22, 38].
- The new and emerging field of coalitional control, which can be seen as an evolution of DMPC where the topology of the control structure can vary with time, has been treated in [39], where an up-to-date literature review is also reported.

Many applications domains of DMPC have been explored, although most of the reported research still makes reference to simulation studies, with only few real applications (mainly laboratory experiments). In view of its nature, DMPC fits very well with the control of large, spatially distributed systems, possibly interconnected through a network. For this reason, power networks, smart grids, and in general distributed energy management systems are the natural domains where DMPC can offer advantages with respect to a centralized solution, see, for instance, [40–44]. Another class of problems where DMPC can have a great potential impact concerns

the management and control of irrigation canals, as discussed in [45]. The coordination of multi-vehicle systems with DMPC has been considered, e.g., in [46, 47]. Finally, applications in other fields are described in [48, 49].

6 Conclusions and Future Perspectives

The research activity in Distributed Model Predictive Control has been intense in the last decade and many methods are nowadays available, see, for instance, the many contributions reported in [4]. In parallel with the development of new algorithms, also the most significant fields of application of DMPC have been clarified. In our opinion, these include networked systems, like power, water, and traffic networks, the coordination of autonomous vehicles and flying systems (drones), the control of very large-scale, weakly coupled, systems. However, there is still a significant gap between research results and real-world applications since most of the DMPC algorithms have only been tested in simulations or with laboratory benchmarks. Among the most promising future research directions, we believe that the reconfigurability of DMPC will be a major topic. To this regard, plug-and-play and coalitional control strategies, possibly driven by external events, will be required to enhance the ability of DMPC to deal with many real control problems and to provide significant improvements with respect to the nowadays adopted control solutions.

References

1. Lunze, J.: Feedback Control of Large-Scale Systems. Series in Systems and Control Engineering, Prentice Hall International, Englewood Cliffs (1992)
2. Šiljak, D.D.: Decentralized control of complex systems. Courier Corporation, North Chelmsford (2011)
3. Scattolini, R.: Architectures for distributed and hierarchical model predictive control - a review. *J. Process Control* **19**, 723–731 (2009)
4. Maestre, J.M., Negenborn, R.R.: Distributed Model Predictive Control Made Easy. Springer, Berlin (2014)
5. Skogestad, S., Postlethwaite, I.: Multivariable Feedback Control: Analysis and Design. Wiley, London (2005)
6. Kariwala, V., Forbes, J.F., Meadows, E.S.: Block relative gain: properties and pairing rules. *Ind. Eng. Chem. Res.* **42**(20), 4564–4574 (2003)
7. Farina, M., Colaneri, P., Scattolini, R.: Block-wise discretization accounting for structural constraints. *Automatica* **49**(11), 3411–3417 (2013)
8. Rawlings, J.B., Mayne, D.Q.: Model Predictive Control: Theory and Design. Nob Hill Publishing, Madison (2009)
9. Alessio, A., Barcelli, D., Bemporad, A.: Decentralized model predictive control of dynamically-coupled linear systems. *J. Process Control* **21**(5), 705–714 (2011)
10. Magni, L., Scattolini, R.: Stabilizing decentralized model predictive control of nonlinear systems. *Automatica* **42**(7), 1231–1236 (2006)
11. Raimondo, D.M., Magni, L., Scattolini, R.: Decentralized MPC of nonlinear system: an input-to-state stability approach. *Int. J. Robust Nonlinear Control* **17**(5), 1651–1667 (2007)

12. Rivero, S., Farina, M., Ferrari-Trecate, G.: Plug-and-play decentralized model predictive control for linear systems. *IEEE Trans. Autom. Control* **58**(10), 2608–2614 (2013)
13. Mayne, D.Q., Seron, M.M., Raković, S.V.: Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica* **41**(2), 219–224 (2005)
14. Raković, S.V., Levine, W.S., Açıkmüşe, A.B.: Elastic tube model predictive control. In: *Proceedings of the 2016 American Control Conference*, pp. 3594–3599 (2016)
15. Rawlings, J.B., Stewart, B.T.: Coordinating multiple optimization-based controllers: new opportunities and challenges. *J. Process Control* **18**, 839–845 (2008)
16. Lopes de Lima, M., Camponogara, E., Limon, D., Muñoz de la Peña, D.: Distributed satisficing MPC. *IEEE Trans. Control Syst. Technol.* **23**(1), 305–312 (2015)
17. Farina, M., Scattolini, R.: Distributed predictive control: a non-cooperative algorithm with neighbor-to-neighbor communication for linear systems. *Automatica* **6**, 1088–1096 (2012)
18. Betti, G., Farina, M., Scattolini, R.: Realization issues, tuning, and testing of a distributed predictive control algorithm. *J. Process Control* **24**(4), 424–434 (2014)
19. Trodden, P., Richards, A.: Distributed model predictive control of linear systems with persistent disturbances. *Int. J. Control* **83**, 1653–1663 (2010)
20. Trodden, P., Richards, A.: Cooperative tube-based distributed MPC for linear uncertain systems coupled via constraints. In: Maestre, J.M., Negenborn, R.R. (eds.) *Distributed Model Predictive Control Made Easy*. Springer, Berlin (2014)
21. Müller, M.A., Reble, M., Allgöwer, F.: Cooperative control of dynamically decoupled systems via distributed model predictive control. *Int. J. Robust Nonlinear Control* **22**, 1376–1397 (2012)
22. Müller, M.A., Allgöwer, F.: Distributed economic MPC: a framework for cooperative control problems. In: *Proceedings of IFAC World Congress*, pp. 1029–1034 (2014)
23. Mesarovic, M.D., Macko, D., Takahara, Y.: *Theory of Hierarchical, Multilevel, Systems*. Academic Press, New York (1970)
24. Giselsson, P., Rantzer, A.: Distributed model predictive control with suboptimality and stability guarantees. In: *Proceedings of the 49th Conference on Decision and Control*, pp. 7272–7277 (2010)
25. Necoara, I., Nedelcu, V., Dumitrache, I.: Parallel and distributed optimization methods for estimation and control in networks. *J. Process Control* **21**, 756–766 (2011)
26. Giselsson, P., Doan, M.D., Keviczky, T., De Schutter, B., Rantzer, A.: Accelerated gradient methods and dual decomposition in distributed model predictive control. *Automatica* **49**, 829–833 (2013)
27. Giselsson, P., Rantzer, A.: On feasibility, stability and performance in distributed model predictive control. *IEEE Trans. Autom. Control* **59**, 1031–1036 (2014)
28. Conte, C., Jones, C.N., Morari, M., Zeilinger, M.N.: Distributed synthesis and stability of cooperative distributed model predictive control for linear systems. *Automatica* **69**, 117–125 (2016)
29. Farina, M., Betti, G., Scattolini, R.: Distributed predictive control of continuous-time systems. *Syst. Control Lett.* **74**, 32–40 (2014)
30. Dunbar, W.B.: Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Trans. Autom. Control* **52**, 1249–1263 (2007)
31. Liu, J., Muñoz de la Peña, D., Christofides, P.D.: Distributed model predictive control of nonlinear process systems. *AIChE J.* **55**, 1171–1184 (2007)
32. Farina, M., Scattolini, R.: An output feedback distributed predictive control algorithm. In: *Proceedings of the 50th IEEE Conference on Decision and Control*, pp. 8139–8144 (2011)
33. Farina, M., Giulioni, L., Betti, G., Scattolini, R.: An approach to distributed predictive control for tracking - theory and applications. *IEEE Trans. Control Syst. Technol.* **22**(4), 1558–1566 (2014)
34. Casavola, A., Garone, E., Tedesco, F.: The distributed command governor approach in a nutshell. In: Maestre, J.M., Negenborn, R.R. (eds.) *Distributed Model Predictive Control Made Easy*. Springer, Berlin (2014)

35. Perizzato, A., Farina, M., Scattolini, R.: Stochastic distributed predictive control of independent systems with coupling constraints. In: IEEE Conference on Decision and Control, pp. 3228–3233 (2014)
36. Farina, M., Giulioni, L., Scattolini, R.: Distributed predictive control of stochastic linear systems with chance constraints. In: American Control Conference, pp. 20–25 (2016)
37. Dai, L., Xia, Y., Gao, Y., Cannon, M.: Distributed stochastic MPC of linear systems with additive uncertainty and coupled probabilistic constraints. *IEEE Trans. Autom. Control* **62**(7), 3474–3481 (2017)
38. Chen, X., Heidarinejad, M., Liu, J., Christofides, P.D.: Distributed economic MPC: application to a nonlinear chemical process network. *J. Process Control* **22**, 689–699 (2012)
39. Fele, F., Maestre, J.M., Camacho, E.F.: Coalitional control: cooperative game theory and control. *IEEE Control Syst. Mag.* **37**, 53–69 (2017)
40. Wang, D., Glavic, M., Wehenkel, L.: Comparison of centralized, distributed and hierarchical model predictive control schemes for electromechanical oscillations damping in large-scale power systems. *Int. J. Electr. Power Energy Syst.* **58**, 32–41 (2014)
41. Ma, M., Chen, H., Liu, X., Allgöwer, F.: Distributed model predictive load frequency control of multi-area interconnected power system. *Int. J. Electr. Power Energy Syst.* **62**, 289–298 (2014)
42. Larsen, G.K.H., van Foreest, N.D., Scherpen, J.M.A.: Distributed MPC applied to a network of households with micro-CHP and heat storage. *IEEE Trans. Smart Grid* **5**, 2106–2114 (2014)
43. del Real, A.J., Arce, A., Bordons, C.: An integrated framework for distributed model predictive control of large-scale power networks. *IEEE Trans. Ind. Inf.* **10**, 197–209 (2014)
44. Scherer, H.F., Pasamontes, M., Guzman, J.L., Alvarez, J.D., Camponogara, E., Normey-Rico, J.E.: Efficient building energy management using distributed model predictive control. *J. Process Control* **24**, 740–749 (2014)
45. Fele, F., Maestre, J.M., Hashemy, S.M., Muñoz de la Peña, D., Camacho, E.F.: Coalitional model predictive control of an irrigation canal. *J. Process Control* **24**, 314–325 (2014)
46. Zhu, M., Martinez, S.: On distributed constrained formation control in operator-vehicle adversarial networks. *Automatica* **49**, 3571–3582 (2013)
47. Farina, M., Perizzato, A., Scattolini, R.: Application of distributed predictive control to motion and coordination problems for unicycle autonomous robots. *Robot. Auton. Syst.* **72**, 248–260 (2015). *Control* **59**, 1439–1453
48. Kirubakaran, V., Radhakrishnan, T.K., Sivakumaran, N.: Distributed multiparametric model predictive control design for a quadruple tank process. *Measurement* **47**, 841–854 (2014)
49. Ferrara, A., Nai Oleari, A., Sacone, S., Siri, S.: Freeways as systems of systems: a distributed model predictive control scheme. *IEEE Syst. J.* **9**, 312–323 (2015)