

D

Distributed Model Predictive Control



Gabriele Pannocchia
University of Pisa, Pisa, Italy

Abstract

Distributed model predictive control refers to a class of predictive control architectures in which a number of local controllers manipulate a subset of inputs to control a subset of outputs (states) composing the overall system. Different levels of communication and (non)cooperation exist, although in general the most compelling properties can be established only for cooperative schemes, those in which all local controllers optimize local inputs to minimize the same plantwide objective function. Starting from state-feedback algorithms for constrained linear systems, extensions are discussed to cover output feedback, reference target tracking, and nonlinear systems. An outlook of future directions is finally presented.

Keywords

Constrained large-scale systems; Cooperative control systems; Interacting dynamical systems

Introduction and Motivations

Large-scale systems (e.g., industrial processing plants, power generation networks, etc.) usually

comprise several interconnected units which may exchange material, energy, and information streams. The overall effectiveness and profitability of such large-scale systems depend strongly on the level of local effectiveness and profitability of each unit but also on the level of interactions among the different units. An overall optimization goal can be achieved by adopting a single *centralized* model predictive control (MPC) system (Rawlings et al. 2017) in which *all* control input trajectories are optimized simultaneously to minimize a *common* objective.

This choice is often avoided for several reasons. When the overall number of inputs and states is very large, a single optimization problem may require computational resources (CPU time, memory, etc.) that are not available and/or compatible with the system's dynamics. Even if these limitations do not hold, it is often the case that organizational reasons require the use of smaller, local controllers, which are easier to coordinate and maintain.

Thus, industrial control systems are often *decentralized*, i.e., the overall system is divided into (possibly mildly coupled) subsystems, and a local controller is designed for each unit disregarding the interactions from/to other subsystems. Depending on the extent of dynamic coupling, it is well-known that the performance of such decentralized systems may be poor and stability properties may be even lost. *Distributed* predictive control architectures arise to meet performance specifications (stability at minimum) similar to centralized predictive

control systems, still retaining the modularity and local character of the optimization problems solved by each controller.

Definitions and Architectures for Constrained Linear Systems

Subsystem Dynamics, Constraints, and Objectives

We start the description of distributed MPC algorithms by considering an overall discrete-time linear time-invariant system in the form:

$$x^+ = Ax + Bu, \quad y = Cx \quad (1)$$

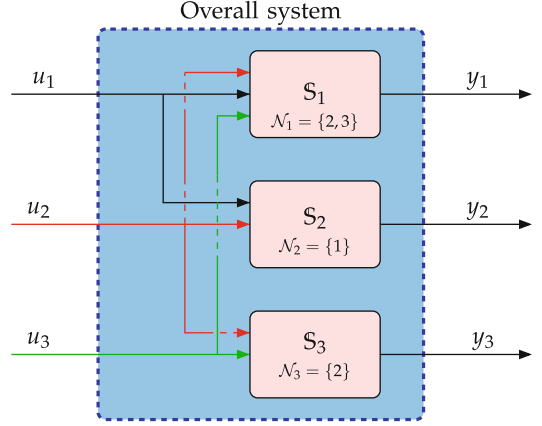
in which $x \in \mathbb{R}^n$ and $x^+ \in \mathbb{R}^n$ are, respectively, the system state at a given time and, at a successor time, $u \in \mathbb{R}^m$ is the input and $y \in \mathbb{R}^p$ is the output.

We consider that the overall system (1) is divided into M subsystems, \mathbb{S}_i , defined by (dis-joint) sets of inputs and outputs (states), and each \mathbb{S}_i is regulated by a *local* MPC. For each \mathbb{S}_i , we denote by $y_i \in \mathbb{R}^{p_i}$ its output, by $x_i \in \mathbb{R}^{n_i}$ its state, and by $u_i \in \mathbb{R}^{m_i}$ the control input computed by the i th MPC. Due to interactions among subsystems, the local output y_i (and state x_i) is affected by control inputs computed by (some) other MPCs. Hence, the dynamics of \mathbb{S}_i can be written as

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

in which \mathcal{N}_i denotes the indices of *neighbors* of \mathbb{S}_i , i.e., the subsystems whose inputs have an influence on the states of \mathbb{S}_i . To clarify the notation, we depict in Fig. 1 the case of three subsystems, with neighbors $\mathcal{N}_1 = \{2, 3\}$, $\mathcal{N}_2 = \{1\}$, and $\mathcal{N}_3 = \{2\}$.

Without loss of generality, we assume that each pair (A_i, B_i) is stabilizable. Moreover, the state of each subsystem x_i is assumed known (to the i th MPC) at each decision time. For each subsystem \mathbb{S}_i , inputs are required to fulfill (hard) constraints:



Distributed Model Predictive Control, Fig. 1
Interconnected systems and neighbors definition

$$u_i \in \mathbb{U}_i, \quad i = 1, \dots, M \quad (3)$$

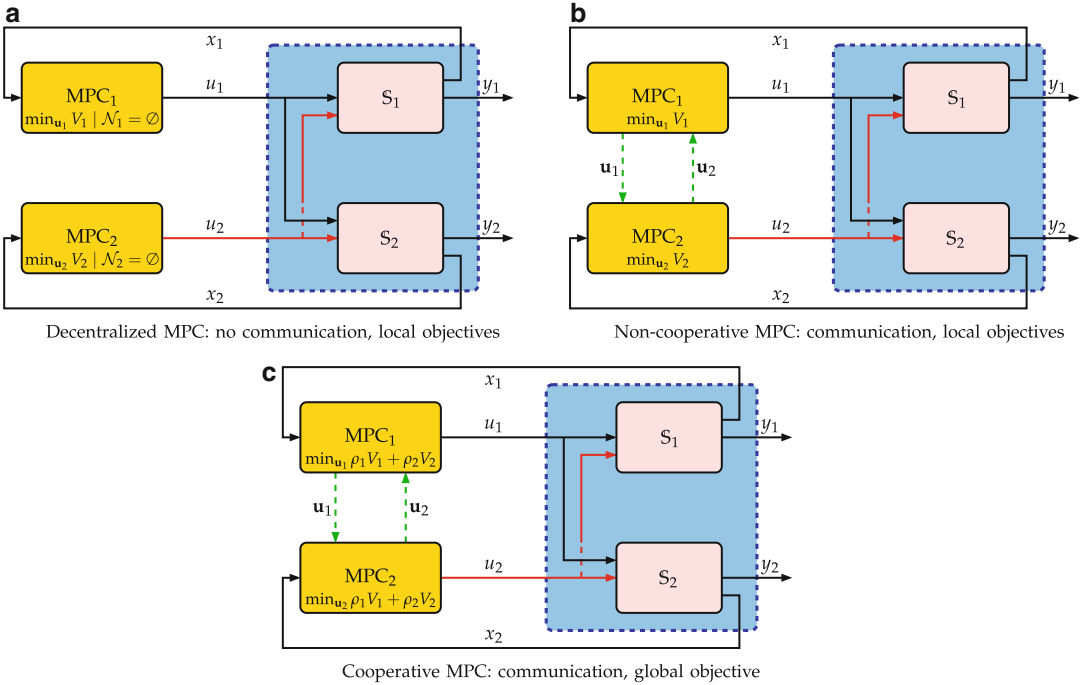
in which \mathbb{U}_i are polyhedrons containing the origin in their interior. Moreover, we consider a quadratic stage cost function $\ell_i(x, u) \triangleq \frac{1}{2}(x' Q_i x + u' R_i u)$ and a terminal cost function $V_{fi}(x) \triangleq \frac{1}{2}x' P_i x$, with $Q_i \in \mathbb{R}^{n_i \times n_i}$, $R_i \in \mathbb{R}^{m_i \times m_i}$, and $P_i \in \mathbb{R}^{n_i \times n_i}$ positive definite. Without loss of generality, let $x_i(0)$ be the state of \mathbb{S}_i at the current decision time. Consequently, the finite-horizon cost function associated with \mathbb{S}_i is given by:

$$V_i(x_i(0), \mathbf{u}_i, \{\mathbf{u}_j\}_{j \in \mathcal{N}_i}) \triangleq \sum_{i=0}^{N-1} \ell_i(x_i(k), u_i(k)) + V_{fi}(x_i(N)) \quad (4)$$

in which $\mathbf{u}_i = (u_i(0), u_i(1), \dots, u_i(N-1))$ is a finite-horizon sequence of control inputs of \mathbb{S}_i and \mathbf{u}_j is similarly defined as a sequence of control inputs of each neighbor $j \in \mathcal{N}_i$. Notice that $V_i(\cdot)$ is a function of neighbors' input sequences, $\{\mathbf{u}_j\}_{j \in \mathcal{N}_i}$, due to the dynamics (2).

Decentralized, Noncooperative, and Cooperative Predictive Control Architectures

Several levels of communications and (non)cooperation can exist among the controllers, as depicted in Fig. 2 for the case of two subsystems.



Distributed Model Predictive Control, Fig. 2 Three distributed control architectures: decentralized MPC, noncooperative MPC, and cooperative MPC

In *decentralized* MPC architectures, interactions among subsystems are neglected by forcing $\mathcal{N}_i = \emptyset$ for all i even if this is not true. That is, the subsystem model used in each local controller, instead of (2), is simply

$$x_i^+ = A_i x_i + B_i u_i, \quad y_i = C_i x_i \quad (5)$$

Therefore, an inherent mismatch exists between the model used by the local controllers (5) and the actual subsystem dynamics (2). Each local MPC solves the following finite-horizon optimal control problem (FHOCP):

$$\mathbb{P}_i^{\text{De}} : \min_{\mathbf{u}_i} V_i(\cdot) \quad \text{s.t. } \mathbf{u}_i \in \mathbb{U}_i^N, \mathcal{N}_i = \emptyset \quad (6)$$

We observe that in this case, $V_i(\cdot)$ depends only on local inputs, \mathbf{u}_i , because it is assumed that $\mathcal{N}_i = \emptyset$. Hence, each \mathbb{P}_i^{De} is solved independently of the neighbors computations, and no iterations are performed. Clearly, depending on the actual level of interactions among subsystems, decentralized MPC architectures can per-

form poorly, namely, being non-stabilizing. Performance certifications are still possible resorting to robust stability theory, i.e., by treating the neglected dynamics $\sum_{j \in \mathcal{N}_i} B_{ij} u_j$ as (bounded) disturbances (Riverso et al. 2013).

In *noncooperative* MPC architectures, the existing interactions among the subsystems are fully taken into account through (2). Given a known value of the neighbors' control input sequences, $\{\mathbf{u}_j\}_{j \in \mathcal{N}_i}$, each local MPC solves the following FHOCP:

$$\mathbb{P}_i^{\text{NCDi}} : \min_{\mathbf{u}_i} V_i(\cdot) \quad \text{s.t. } \mathbf{u}_i \in \mathbb{U}_i^N \quad (7)$$

The obtained solution can be exchanged with the other local controllers to update the assumed neighbors' control input sequences, and iterations can be performed. We observe that this approach is noncooperative because local controllers try to optimize different, possibly competing, objectives. In general, no convergence is guaranteed in noncooperative iterations, and when this scheme converges, it leads to a so-called Nash equilibrium. However, the achieved local control inputs

do not have proven stability properties (Rawlings et al. 2017, §6.2.3). To ensure closed-loop stability, variants can be formulated by including a sequential solution of local MPC problems; exploiting the notion (if any) of an auxiliary stabilizing decentralized control law, non-iterative noncooperative schemes are also proposed, in which stability guarantees are provided by ensuring a decrease of a centralized Lyapunov function at each decision time.

Finally, in *cooperative* MPC architectures, each local controller optimizes a common (plantwide) objective:

$$V(x(0), \mathbf{u}) \triangleq \sum_{i=1}^M \rho_i V_i(x_i(0), \mathbf{u}_i, \{\mathbf{u}_j\}_{j \in \mathcal{N}_i^c}) \quad (8)$$

in which $\rho_i > 0$, for all i , are given scalar weights and $\mathbf{u} \triangleq (\mathbf{u}_1, \dots, \mathbf{u}_M)$ is the overall control sequence. In particular, given a known value of other subsystems' control input sequences, $\{\mathbf{u}_j\}_{j \neq i}$, each local MPC solves the following FHOC:

$$\mathbb{P}_i^{\text{CDi}} : \min_{\mathbf{u}_i} V(\cdot) \quad \text{s.t. } \mathbf{u}_i \in \mathbb{U}_i^N \quad (9)$$

As in noncooperative schemes, the obtained solution can be exchanged with the other local controllers, and further iterations can be performed. Notice that in $\mathbb{P}_i^{\text{CDi}}$, the (possible) implications of the local control sequence \mathbf{u}_i to all other subsystems' objectives, $V_j(\cdot)$ with $j \neq i$, are taken into account, as well as the effect of the neighbors' sequences $\{\mathbf{u}_j\}_{j \in \mathcal{N}_i^c}$ on the local state evolution through (2). Clearly, this approach is termed cooperative because all controllers compute *local* inputs to minimize a *global* objective. Convergence of cooperative iterations is guaranteed, and under suitable assumptions the converged solution is the centralized Pareto-optimal solution (Rawlings et al. 2017, §6.2.4). Furthermore, the achieved local control inputs have proven stabilizing properties (Stewart et al. 2010). Variants are also proposed in which each controller still optimizes a local objective, but cooperative iterations are performed to ensure a decrease of the global objective at each decision time (Maestre et al. 2011).

Cooperative Distributed MPC

Cooperative schemes are preferable over noncooperative schemes from many points of view, namely, in terms of superior theoretical guarantees and no larger computational requirements. In this section we focus on a prototype cooperative distributed MPC algorithm adapted from Stewart et al. (2010), highlighting the required computations and discussing the associated theoretical properties and guarantees.

Basic Algorithm

We present in Algorithm 1 a streamlined description of a cooperative distributed MPC algorithm, in which each local controller solves $\mathbb{P}_i^{\text{CDi}}$, given a previously computed value of all other subsystems' input sequences. For each local controller, the new iterate is defined as a convex combination of the newly computed solution with the previous iteration. A relative tolerance is defined, so that cooperative iterations stop when all local controllers have computed a new iterate sufficiently close to the previous one. A maximum number of cooperative iterations can also be defined, so that a finite bound on the execution time can be established.

Algorithm 1 (*Cooperative MPC*). **Require:**

Overall warm start $\mathbf{u}^0 \triangleq (\mathbf{u}_1^0, \dots, \mathbf{u}_M^0)$, convex step weights $w_i > 0$, s.t. $\sum_{i=1}^M w_i = 1$, relative tolerance parameter $\epsilon > 0$, maximum cooperative iterations c_{\max}

-
- 1: Initialize: $c \leftarrow 0$ and $e_i \leftarrow 2\epsilon$ for $i = 1, \dots, M$.
 - 2: **while** ($c < c_{\max}$) **and** ($\exists i | e_i > \epsilon$) **do**
 - 3: $c \leftarrow c + 1$.
 - 4: **for** $i = 1$ **to** M **do**
 - 5: Solve $\mathbb{P}_i^{\text{CDi}}$ in (9) obtaining \mathbf{u}_i^* .
 - 6: **end for**
 - 7: **for** $i = 1$ **to** M **do**
 - 8: Define new iterate: $\mathbf{u}_i^c \triangleq w_i \mathbf{u}_i^* + (1 - w_i) \mathbf{u}_i^{c-1}$.
 - 9: Compute convergence error: $e_i \triangleq \frac{\|\mathbf{u}_i^c - \mathbf{u}_i^{c-1}\|}{\|\mathbf{u}_i^{c-1}\|}$.
 - 10: **end for**
 - 11: **end while**
 - 12: **return** Overall solution: $\mathbf{u}^c \triangleq (\mathbf{u}_1^c, \dots, \mathbf{u}_M^c)$.
-

We observe that Step 8 implicitly defines the new overall iterate as a convex combination of the

overall solutions *achieved* by each controller, that is,

$$\mathbf{u}^c = \sum_{i=1}^M w_i (\mathbf{u}_1^{c-1}, \dots, \mathbf{u}_i^*, \dots, \mathbf{u}_M^{c-1}) \quad (10)$$

It is also important to observe that Steps 5, 8, and 9 are performed separately by each controller.

Properties

The basic cooperative MPC described in Algorithm 1 enjoys several nice theoretical and practical properties, as detailed (Rawlings et al. 2017, §6.3.1):

1. *Feasibility of each iterate:* $\mathbf{u}_i^{c-1} \in \mathbb{U}_i^N$ implies $\mathbf{u}_i^c \in \mathbb{U}_i^N$, for all $i = 1, \dots, M$ and $c \in \mathbb{I}_{>0}$.
2. *Cost decrease at each iteration:* $V(x(0), \mathbf{u}^c) \leq V(x(0), \mathbf{u}^{c-1})$ for all $c \in \mathbb{I}_{>0}$.
3. *Cost convergence to the centralized optimum:* $\lim_{c \rightarrow \infty} V(x(0), \mathbf{u}^c) = \min_{\mathbf{u} \in \mathbb{U}^N} V(x(0), \mathbf{u})$, in which $\mathbb{U} \triangleq \mathbb{U}_1 \times \dots \times \mathbb{U}_M$.

Resorting to suboptimal MPC theory, the above properties (1) and (2) can be exploited to show that the origin of closed-loop system

$$x^+ = Ax + B\kappa^c(x), \text{ with } \kappa^c(x) \triangleq u^c(0) \quad (11)$$

is *exponentially stable* for any finite $c \in \mathbb{I}_{>0}$. This result is of paramount (practical and theoretical) importance because it ensures closed-loop stability using cooperative distributed MPC with any finite number of cooperative iterations. As in centralized MPC based on the solution of a FHOCP (Rawlings et al. 2017, §2.4.2), particular care of the terminal cost function $V_{fi}(\cdot)$ is necessary, possibly in conjunction with a terminal constraint $x_i(N) \in \mathbb{X}_{fi}$. Several options can be adopted as discussed, e.g., in Stewart et al. (2010, 2011).

Moreover, the results in Pannocchia et al. (2011) can be used to show *inherent robust stability* to system's disturbances and measurement errors. Therefore, we can confidently state that well-designed distributed cooperative MPC and centralized MPC algorithms share the *same guarantees* in terms of stability and robustness.

Complementary Aspects

We discuss in this section a number of complementary aspects of distributed MPC algorithms, omitting technical details for the sake of space.

Coupled Input Constraints and State Constraints

Convergence of the solution of cooperative distributed MPC toward the centralized (global) optimum holds when input constraints are in the form of (3), i.e., when no constraints involve inputs of different subsystems. Sometimes this assumption fails to hold, e.g., when several units share a common utility resource, that is, in addition to (3) some constraints involve inputs of more than one unit. In this situation, it is possible that Algorithm 1 remains stuck at a fixed point, without improving the cost, even if it is still away from the centralized optimum (Rawlings et al. 2017, §6.3.2). It is important to point out that this situation is harmless from a closed-loop stability and robustness point of view. However, the degree of suboptimality in comparison with centralized MPC could be undesired from a performance point of view. To overcome this situation, a slightly different partitioning of the overall inputs into non-disjoint sets can be adopted (Stewart et al. 2010).

Similarly the presence of state constraints, even in decentralized form $x_i \in \mathbb{X}_i$ (with $i = 1, \dots, M$), can prevent convergence of a cooperative algorithm toward the centralized optimum. It is also important to point out that the local MPC controlling \mathcal{S}_i needs to consider in the optimal control problem, besides local state constraints $x_i \in \mathbb{X}_i$ and also state constraints of all other subsystems \mathcal{S}_j such that $i \in \mathcal{N}_j$. This ensures feasibility of each iterate and cost reduction; hence closed-loop stability (and robustness) can be established.

Another route to address the presence of coupling constraints is that based on suitable decomposition methods for distributed optimization, such as the dual decomposition and alternative direction method of multipliers (Doan et al. 2011; Farokhi et al. 2014).

Parsimonious Local System Representations

In cooperative distributed algorithms, each local optimization problem (9) is solved considering the evolution of the overall system state over the prediction horizon, so that the centralized cost function can be optimized by each local controller. Depending on the interconnections among the subsystems, this approach could be simplified by considering a subset of overall state dynamics, without losing the global optimality guarantees, as proposed in Razzanelli and Pannocchia (2017) and briefly reviewed.

From graph theory, we define the *outlet star* of each local subsystem \mathbb{S}_i , as the set of subsystems \mathbb{S}_j of which \mathbb{S}_i is neighbor, i.e., $\mathcal{O}_i = \{j \mid i \in \mathcal{N}_j\}$. Thus, the augmented state dynamics that should be considered by \mathbb{S}_i is composed by (2) and

$$\begin{aligned} x_j^+ &= A_j x_j + B_{ji} u_i + B_j u_j + \sum_{k \in \mathcal{N}_j \setminus i} B_{jk} u_k, \\ y_j &= C_j x_j \quad \forall j \in \mathcal{O}_i \end{aligned} \quad (12)$$

Thus, the FHOCP (9) can be simplified to the following, more parsimonious, problem:

$$\mathbb{P}_i^{\text{PaCDi}} : \min_{\mathbf{u}_i} V_i(\cdot) + \sum_{j \in \mathcal{O}_i} V_j(\cdot) \quad \text{s.t. } \mathbf{u}_i \in \mathbb{U}_i^N \quad (13)$$

obtaining the same solution with significant computational savings (Razzanelli and Pannocchia 2017).

Output Feedback and Offset-Free Tracking

When the subsystem state cannot be directly measured, each local controller can use a local state estimator, namely, a Kalman filter (or Luenberger observer). Assuming that the pair (A_i, C_i) is detectable, the subsystem state estimate evolves as follows:

$$\hat{x}_i^+ = A_i \hat{x}_i + B_i u_i + \sum_{j \in \mathcal{N}_i} B_{ij} u_j + L_i (y_i - C_i \hat{x}_i) \quad (14)$$

in which $L_i \in \mathbb{R}^{n_i \times p_i}$ is the local Kalman predictor gain, chosen such that the matrix $(A_i - L_i C_i)$ is Schur. Stability of the closed-loop origin can be

still established using minor variations (Rawlings et al. 2017, §6.3.3).

When offset-free control is sought, each local MPC can be equipped with an integrating disturbance model similarly to centralized offset-free MPC algorithms (Pannocchia 2015). Given the current estimate of the subsystem state and disturbance, a target calculation problem is solved to compute the state and input equilibrium pair such that (a subset of) output variables correspond to given set points. Such a target calculation problem can be performed in a centralized fashion or in a distributed manner (Razzanelli and Pannocchia 2017). Furthermore, the target calculation problem can be embedded into the FHOCP (Ferramosca et al. 2013; Razzanelli and Pannocchia 2017).

Distributed Control for Nonlinear Systems

Several nonlinear distributed MPC algorithms have been recently proposed (Liu et al. 2009; Stewart et al. 2011). Some schemes require the presence of a coordinator, thus introducing a hierarchical structure (Scattolini 2009). In Stewart et al. (2011), instead, a cooperative distributed MPC architecture similar to the one discussed in the previous section has been proposed for nonlinear systems. Each local controller considers the following subsystem model:

$$\begin{aligned} x_i^+ &= f_i(x_i, u_i, u_j), \quad \text{with } j \in \mathcal{N}_i \\ y_i &= h_i(x_i) \end{aligned} \quad (15)$$

A problem (formally) identical to $\mathbb{P}_i^{\text{CDi}}$ in (9) is solved by each controller, and cooperative iterations are performed. However, non-convexity of $\mathbb{P}_i^{\text{CDi}}$ can make a convex combination step similar to Step 8 in Algorithm 1 not necessarily a cost improvement. As a workaround in such cases, Stewart et al. (2011) propose deleting the least effective control sequence computed by a local controller (repeating this deletion if necessary). In this way it is possible to show a monotonic decrease of the cost function at each cooperative iteration (Rawlings et al. 2017, §6.5).

Summary and Future Directions

We presented the basics and foundations of distributed model predictive control (DMPC) schemes, which prove useful and effective in the control of large-scale systems for which a single centralized predictive controller is not regarded as a possible or desirable solution, e.g., due to organizational requirements and/or computational limitations. In DMPCs, the overall controlled system is organized into a number of subsystems, in general featuring some dynamic couplings, and for each subsystem a local MPC is implemented.

Different flavors of communication and cooperation among the local controllers can be chosen by the designer, ranging from *decentralized* to *cooperative* schemes. In cooperative DMPC algorithms, the dynamic interactions among the subsystems are fully taken into account, with limited communication overheads, and the same overall objective can be optimized by each local controller. When cooperative iterations are performed upon convergence, such DMPC algorithms achieve the same global minimum control sequence as that of the centralized MPC. Termination prior to convergence does not hinder stability and robustness guarantees.

In this contribution, after discussing an overview on possible communication and cooperation schemes, we addressed the design of a state-feedback and distributed MPC algorithm for linear systems subject to input constraints, with convergence and stability guarantees. Then, we discussed various extensions to coupled input constraints and state constraints, output feedback, reference target tracking, and nonlinear systems.

The research on DMPC algorithms has been extensive during the last decade, and some excellent review papers have been recently made available (Christofides et al. 2013; Scattolini 2009). Still, we expect DMPC to attract research efforts in various directions, as briefly discussed:

- *Nonlinear DMPC* algorithms (Liu et al. 2009; Stewart et al. 2011; Hours and Jones 2015)

will require improvements in terms of global optimum goals.

- *Economic DMPC* and *tracking DMPC* (Ferramosca et al. 2013; Razzanelli and Pannocchia 2017; Köhler et al. 2018) will replace current formulations designed for regulation around the origin.
- *Reconfigurability*, e.g., addition/deletion of new local subsystems and controllers, is an ongoing topic, which finds particular applicability in the context of smart-grid systems (Molzahn et al. 2017). Preliminary results, discussed in Rivero et al. (2013) for decentralized control structures, have been also extended to distributed nonlinear systems (Rivero et al. 2016).
- *Constrained distributed estimation*, preliminarily addressed in Farina et al. (2012), is drawing attention (Yin and Liu 2017), although further insights are needed to bridge the gap between constrained estimation and control algorithms.
- Specific *distributed optimization algorithms* tailored to DMPC problems (Doan et al. 2011; Farokhi et al. 2014; Molzahn et al. 2017) will increase the effectiveness of DMPC algorithms.

Cross-References

- ▶ [Cooperative Solutions to Dynamic Games](#)
- ▶ [Nominal Model-Predictive Control](#)
- ▶ [Optimization Algorithms for Model Predictive Control](#)
- ▶ [Tracking Model Predictive Control](#)

Recommended Reading

General overviews on DMPC can be found in Christofides et al. (2013), Rawlings et al. (2017), and Scattolini (2009). DMPC algorithms for linear systems are discussed in Ferramosca et al. (2013), Rivero et al. (2013), Stewart et al. (2010), Maestre et al. (2011), and Razzanelli and Pannocchia (2017) and for nonlinear systems in

Farina et al. (2012), Liu et al. (2009), and Stewart et al. (2011).

Bibliography

- Christofides PD, Scattolini R, Muñoz de la Peña D, Liu J (2013) Distributed model predictive control: a tutorial review and future research directions. *Comput Chem Eng* 51:21–41
- Doan MD, Keviczky T, De Schutter B (2011) An iterative scheme for distributed model predictive control using Fenchel's duality. *J Process Control* 21:746–755
- Farina M, Ferrari-Trecate G, Scattolini R (2012) Distributed moving horizon estimation for nonlinear constrained systems. *Int J Robust Nonlinear Control* 22:123–143
- Farokhi F, Shames I, Johansson KH (2014) Distributed MPC via dual decomposition and alternative direction method of multipliers. In: Maestre J, Negenborn R (eds) *Distributed model predictive control made easy. Intelligent systems, control and automation: science and engineering*, vol 69. Springer, Dordrecht
- Ferramosca A, Limon D, Alvarado I, Camacho EF (2013) Cooperative distributed MPC for tracking. *Automatica* 49:906–914
- Hours JH, Jones CN (2015) A parametric nonconvex decomposition algorithm for real-time and distributed NMPC. *IEEE Trans Autom Control* 61(2):287–302
- Köhler PN, Müller M, Allgöwer F (2018) A distributed economic MPC framework for cooperative control under conflicting objectives. *Automatica* 96:368–379
- Liu J, Muñoz de la Peña D, Christofides PD (2009) Distributed model predictive control of nonlinear process systems. *AIChE J* 55(5):1171–1184
- Maestre JM, Muñoz de la Peña D, Camacho EF (2011) Distributed model predictive control based on a cooperative game. *Optim Control Appl Methods* 32:153–176
- Molzahn DK, Dörfler F, Sandberg H, Low SH, Chakrabarti S, Baldick R, Lavaei J (2017) A survey of distributed optimization and control algorithms for electric power systems. *IEEE Trans Smart Grid* 8(6):2941–2962
- Pannocchia G (2015) Offset-free tracking MPC: a tutorial review and comparison of different formulations. In: 2015 European control conference (ECC), pp 527–532
- Pannocchia G, Rawlings JB, Wright SJ (2011) Conditions under which suboptimal nonlinear MPC is inherently robust. *Syst Control Lett* 60:747–755
- Rawlings JB, Mayne DQ, Mayne DQ, Diehl MM (2017) *Model predictive control: theory, computation, and design*, 2nd edn. Nob Hill Publishing, Madison
- Razzanelli M, Pannocchia G (2017) Parsimonious cooperative distributed MPC algorithms for offset-free tracking. *J Process Control* 60:1–13
- Riverso S, Farina M, Ferrari-Trecate G (2013) Plug-and-play decentralized model predictive control for linear systems. *IEEE Trans Autom Control* 58:2608–2614
- Riverso S, Boem F, Ferrari-Trecate G, Parisini T (2016) Plug-and-play fault detection and control-reconfiguration for a class of nonlinear large-scale constrained systems. *IEEE Trans Autom Control* 61:3963–3978
- Scattolini R (2009) A survey on hierarchical and distributed model predictive control. *J Process Control* 19:723–731
- Stewart BT, Venkat AN, Rawlings JB, Wright SJ, Pannocchia G (2010) Cooperative distributed model predictive control. *Syst Control Lett* 59:460–469
- Stewart BT, Wright SJ, Rawlings JB (2011) Cooperative distributed model predictive control for nonlinear systems. *J Process Control* 21:698–704
- Yin X, Liu J (2017) Distributed moving horizon state estimation of two-time-scale nonlinear systems. *Automatica* 79(5):152–161