Chapter 5
Decentralized Model Predictive Control

Alberto Bemporad and Davide Barcelli

Abstract. Decentralized and distributed model predictive control (DMPC) addresses the problem of controlling a multivariable dynamical process, composed by several interacting subsystems and subject to constraints, in a computation and communication efficient way. Compared to a centralized MPC setup, where a global optimal control problem must be solved on-line with respect to all actuator commands given the entire set of states, in DMPC the control problem is divided into a set of local MPCs of smaller size, that cooperate by communicating each other a certain information set, such as local state measurements, local decisions, optimal local predictions. Each controller is based on a partial (local) model of the overall dynamics, possibly neglecting existing dynamical interactions. The global performance objective is suitably mapped into a local objective for each of the local MPC problems.

This chapter surveys some of the main contributions to DMPC, with an emphasis on a method developed by the authors, by illustrating the ideas on motivating examples. Some novel ideas to address the problem of hierarchical MPC design are also included in the chapter.

5.1 Introduction

Most of the procedures for analyzing and controlling dynamical systems developed over the last decades rest on the common presupposition of *centrality*. Centrality means that all the information available about the system is collected at a single location, where all the calculations based on such information are executed. Information includes both *a priori* information about the dynamical model of the system available off-line, and *a posteriori* information about the system response gathered

Alberto Bemporad
Department of Mechanical and Structural Engineering, University of Trento
e-mail: bemporad@ing.unitn.it

Davide Barcelli
Department of Information Engineering, University of Siena, Italy
e-mail: barcelli@dii.unisi.it

A. Bemporad, M. Heemels, and M. Johansson: Networked Control Systems, LNCIS 406, pp. 149–178.
springerlink.com

by different sensors on-line. When considering large-scale systems the presupposition of centrality fails because of the lack of a centralized information-gathering system or of centralized computing capabilities. Typical examples of such systems are power networks, water networks, urban traffic networks, cooperating vehicles, digital cellular networks, flexible manufacturing networks, supply chains, complex structures in civil engineering,and many others. In such systems the centrality assumption often fails because of geographical separation of components (spatial distribution), as the costs and the reliability of communication links cannot be neglected. Moreover, technological advances and reduced cost of microprocessors provide a new force for distributed computation. Hence the current trend for *decentralized* decision making, distributed computations, and hierarchical control.

Several new challenges arise when addressing a decentralized setting, where most of the existing analysis and control design methodologies cannot be directly applied. In a distributed control system which employs decentralized control techniques there are several local control stations, where each controller observes only local outputs and only controls local inputs. Besides advantages in controller implementation (namely reduced and parallel computations, reduced communications), a great advantage of decentralization is maintenance: while certain parts of the overall process are interrupted, the remaining parts keep operating in closed-loop with their local controllers, without the need of stopping the overall process as in case of centralized control. Moreover, a partial re-design of the process does not necessarily imply a complete re-design of the controller, as it would instead in case of centralized control. However, all the controllers are involved in controlling the same large-scale process, and is therefore of paramount importance to determine conditions under which there exists a set of appropriate local feedback control laws stabilizing the entire system.

Ideas for decentralizing and hierarchically organizing the control actions in industrial automation systems date back to the 70's [37, 26, 27, 31, 11], but were mainly limited to the analysis of stability of decentralized linear control of interconnected subsystems, so the interest faded. Since the late 90's, because of the advances in computation techniques like convex optimization, the interest in decentralized control raised again [14, 29], and convex formulations were developed, although limited to special classes of systems such as spatially invariant systems [4]. Decentralized control and estimation schemes based on distributed convex optimization ideas have been proposed recently in [30, 20] based on Lagrangean relaxations. Here global solutions can be achieved after iterating a series of local computations and inter-agent communications.

Large-scale multi-variable control problems, such as those arising in the process industries, are often dealt with model predictive control (MPC) techniques. In MPC the control problem is formulated as an optimization one, where many different (and possibly conflicting) goals are easily formalized and state and control constraints can be included. Many results are nowadays available concerning stability and robustness of MPC, see *e.g.* [24]. However, centralized MPC is often unsuitable for control of large-scale networked systems, mainly due to lack of scalability and to maintenance issues of global models. In view of the above considerations, it is

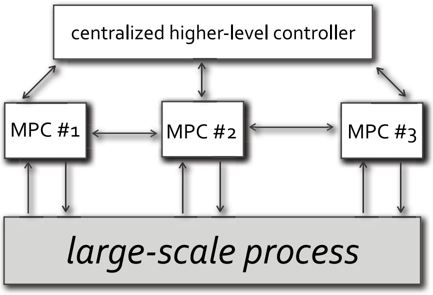


Fig. 5.1 Hierarchical and decentralized/distributed model predictive control of a large-scale process

then natural to look for decentralized or for distributed MPC (DMPC) algorithms, in which the original large-size optimization problem is replaced by a number of smaller and easily tractable ones that work iteratively and cooperatively towards achieving a common, system-wide control objective.

Even though there is not a universal agreement on the distinction between "decentralized" and "distributed", the main difference between the two terms depends on the type of information exchange:

- *decentralized MPC*: Control agents take control decisions independently on each other. Information exchange (such as measurements and previous control decisions) is only allowed before and after the decision making process. There is no negotiation between agents during the decision process. The time needed to decide the control action is not affected by communication issues, such as network delays and loss of packets.
- *distributed MPC*: An exchange of candidate control decisions may also happen during the decision making process, and iterated until an agreement is reached among the different local controllers, in accordance with a given stopping criterion.

In DMPC M subproblems are solved, each one assigned to a different control agent, instead of a single centralized problem. The goal of the decomposition is twofold: first, each subproblem is much smaller than the overall problem (that is, each subproblem has far fewer decision variables and constraints than the centralized one), and second, each subproblem is coupled to only a few other subproblems (that is, it shares variables with only a limited number other subproblems). Although decentralizing the MPC problem may lead to a deterioration of the overall closed-loop performance because of the suboptimality of the resulting control actions, besides computation and communication benefits there are also important operational benefits in using DMPC solutions. For instance local maintenance can be carried out by only stopping the corresponding local MPC controller, while in a centralized MPC approach the whole process should be suspended.

A DMPC control layer is often interacting with a higher-level control layer in a hierarchical arrangement, as depicted in Figure 5.1. The goal of the higher layer is to possibly adjust set-points and constraint specifications to the DMPC layer, based on a *global* (possibly less detailed) model of the entire system. Because of its general overview of the entire process, such a centralized decision layer allows one to reach levels of coordination and performance optimization otherwise very difficult (if not impossible) using a decentralized or distributed action. For a recent survey on decentralized, distributed and hierarchical model predictive control architectures, the reader is referred to the recent survey paper [32].

In a typical DMPC framework the steps performed by the local controllers at each control instant are the following: (*i*) measure local variables and update state estimates, (*ii*) solve the local receding-horizon control problem, (*iii*) apply the control signal for the current instant, (*iv*) exchange information with other controllers. Along with the benefits of a decentralized design, there are some inherent issues that one must face in DMPC: ensuring the asymptotic stability of the overall system, ensure the feasibility of global constraints, quantify the loss of performance with respect to centralized MPC.

5.2 Model Predictive Control

In this section we review the basic setup of linear model predictive control. Consider the problem of regulating the discrete-time linear time-invariant system

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \tag{5.1}$$

to the origin while fulfilling the constraints

$$u_{\min} \leq u(t) \leq u_{\max} \tag{5.2}$$

at all time instants $t \in \mathbb{Z}_{0+}$ where $\mathbb{Z}_{0+}$ is the set of nonnegative integers, $x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$ are the state, input, and output vectors, respectively, and the pair (A,B) is stabilizable. In (5.2) the constraints should be interpreted component-wise and we assume $u_{min} < 0 < u_{max}$.

MPC solves such a constrained regulation problem as described below. At each time t, given the state vector $x(t)$, the following finite-horizon optimal control problem

$$V(x(t)) = \min_{U} \; x'_{t+N} P x_{t+N} + \sum_{k=0}^{N-1} x'_k Q x_k + u'_k R u_k \tag{5.3a}$$

$$\text{s.t.} \; x_{k+1} = Ax_k + Bu_k, \; k = 0, \ldots, N-1 \tag{5.3b}$$

$$y_k = Cx_k, \; k = 0, \ldots, N \tag{5.3c}$$

$$x_0 = x(t) \tag{5.3d}$$

$$u_{\min} \leq u_k \leq u_{\max}, \; k = 0, \ldots, N_u - 1 \tag{5.3e}$$

$$u_k = Kx_k, \; k = N_u, \ldots, N-1 \tag{5.3f}$$

Table 5.1 Classification of existing DMPC approaches.

acronym	submodels	constraints	intersampling iterations	broadcast predictions	state constraints	stability constraints	references
ABB	coupled	local inputs	no	no	no	none	[2, 3, 5, 1]
VRW	coupled	local inputs	yes	no	no	none	[35, 34]
MD	coupled	local inputs	yes	yes	no	none	[25]
DM	decoupled	local inputs	no	yes	yes	compatibility	[15]
KBB	decoupled		no	yes	yes	none	[21]
JK	coupled	local inputs	no	yes	yes	compatibility	[17, 12, 18]

is solved, where $U \triangleq \{u_0, \dots, u_{N_u-1}\}$ is the sequence of future input moves, x_k denotes the predicted state vector at time $t+k$, obtained by applying the input sequence $u_0, \dots, u_{k-1}$ to model (5.1), starting from $x(t)$. In (5.3) $N > 0$ is the prediction horizon, $N_u \leq N-1$ is the input horizon, $Q = Q' \geq 0$, $R = R' > 0$, $P = P' \geq 0$ are square weight matrices defining the performance index, and K is some terminal feedback gain. As we will discuss below, P, K are chosen in order to ensure closed-loop stability of the overall process.

Problem (5.3) can be recast as a quadratic programming (QP) problem (see *e.g.* [24, 9]), whose solution $U^*(x(t)) \triangleq \{u_0^* \ \dots \ u_{N_u-1}^*\}$ is a sequence of optimal control inputs. Only the first input

$$u(t) = u_0^* \tag{5.4}$$

is actually applied to system (5.1), as the optimization problem (5.3) is repeated at time $t+1$, based on the new state $x(t+1)$ (for this reason, the MPC strategy is often referred to as *receding horizon* control). The MPC algorithm (5.3)-(5.4) requires that all the n components of the state vector $x(t)$ are collected in a (possibly remote) central unit, where a quadratic program with mN_u decision variables needs to be solved and the solution broadcasted to the m actuators. As mentioned in the introduction, such a centralized MPC approach may be inappropriate for control of large-scale systems, and it is therefore natural to look for decentralized or distributed MPC (DMPC) algorithms.

5.3 Existing Approaches to DMPC

A few contributions have appeared in recent years in the context of DMPC, mainly motivated by applications of decentralized control of cooperating air vehicles [10, 28, 22]. We review in this section some of the main contributions on DMPC, summarized in Table 5.1, that have appeared in the scientific literature. An application of some of the results surveyed in this chapter in a problem of distributed control of power networks with comparisons among DMPC approaches is reported in [13].

In the following sections, we denote by M be the number of local MPC controllers that we want to design, for example $M = m$ in case each individual actuator is governed by its own local MPC controller.

5.3.1 DMPC Approach of Alessio, Barcelli, and Bemporad

In [2, 3, 5, 1] a decentralized MPC design approach for possibly dynamically coupled processes was proposed. A (partial) decoupling assumption only appears in the *prediction* models used by different MPC controllers. The chosen degree of decoupling represents a tuning knob of the approach. Sufficient criteria for analyzing the asymptotic stability of the process model in closed loop with the set of decentralized MPC controllers are provided. If such conditions are not verified, then the structure of decentralization should be modified by augmenting the level of dynamical coupling of the prediction submodels, increasing consequently the number and type of exchanged information about state measurements among the MPC controllers. Following such stability criteria, a hierarchical scheme was proposed to change the decentralization structure on-line by a supervisory scheme without destabilizing the system. Moreover, to cope with the case of a non-ideal communication channel among neighboring MPC controllers, sufficient conditions for ensuring closed-loop stability of the overall closed-loop system when packets containing state measurements may be lost were given. We review here the main ingredients and results of this approach.

5.3.1.1 Decentralized Prediction Models

Consider again process model (5.1). Matrices A, B may have a certain number of zero or negligible components corresponding to a partial dynamical decoupling of the process, especially in the case of large-scale systems, or even be block diagonal in case of total dynamical decoupling. This is the case for instance of independent moving agents each one having its own dynamics and only coupled by a global performance index.

For all $i = 1, \ldots, M$, we define $x^i \in \mathbb{R}^{n_i}$ as the vector collecting a subset $\mathscr{I}_{xi} \subseteq \{1, \ldots, n\}$ of the state components,

$$x^i = W_i' x = \begin{bmatrix} x_1^i \\ \vdots \\ x_{n_i}^i \end{bmatrix} \in \mathbb{R}^{n_i} \tag{5.5a}$$

where $W_i \in \mathbb{R}^{n \times n_i}$ collects the n_i columns of the identity matrix of order n corresponding to the indices in $\mathscr{I}_{xi}$, and, similarly,

$$u^i = Z_i' u = \begin{bmatrix} u_1^i \\ \vdots \\ u_{m_i}^i \end{bmatrix} \in \mathbb{R}^{m_i} \tag{5.5b}$$

as the vector of input signals tackled by the i-th controller, where $Z_i \in \mathbb{R}^{m \times m_i}$ collects m_i columns of the identity matrix of order m corresponding to the set of indices $\mathscr{I}_{ui} \subseteq \{1, \ldots, m\}$. Note that

$$W_i' W_i = I_{n_i},\ Z_i' Z_i = I_{m_i},\ \forall i = 1, \ldots, M \tag{5.6}$$

where $I_{(\cdot)}$ denotes the identity matrix of order $(\cdot)$. By definition of x^i in (5.5a) we obtain

$$x^i(t+1) = W_i' x(t+1) = W_i' A x(t) + W_i' B u(t) \tag{5.7}$$

An *approximation* of (5.1) is obtained by changing $W_i' A$ in (5.7) into $W_i' A W_i W_i'$ and $W_i' B$ into $W_i' B Z_i Z_i'$, therefore getting the new prediction reduced order model

$$x^i(t+1) = A_i x^i(t) + B_i u^i(t) \tag{5.8}$$

where matrices $A_i = W_i' A W_i \in \mathbb{R}^{n_i \times n_i}$ and $B_i = W_i' B Z_i \in \mathbb{R}^{m_i \times m_i}$ are submatrices of the original A and B matrices, respectively, describing in a possibly approximate way the evolution of the states of subsystem #i.

The size (n_i, m_i) of model (5.8) in general will be much smaller than the size (n, m) of the overall process model (5.1). The choice of the pair (W_i, Z_i) of *decoupling matrices* (and, consequently, of the dimensions n_i, m_i of each submodel) is a tuning knob of the DMPC procedure proposed in the sequel of the paper.

We want to design a controller for each set of moves u^i according to prediction model (5.8) and based on feedback from x^i, for all $i = 1, \ldots, M$. Note that in general different input vectors u^i, u^j may share common components. To avoid ambiguities on the control action to be commanded to the process, we impose that only a subset $\mathscr{I}_{ui}^{\#} \subseteq \mathscr{I}_{ui}$ of input signals computed by controller #i is actually applied to the process, with the following conditions

$$\bigcup_{i=1}^{M} I_{ui}^{\#} = \{1, \ldots, m\} \tag{5.9a}$$

$$I_{ui}^{\#} \cap I_{uj}^{\#} = \emptyset,\ \forall i, j = 1, \ldots, M,\ i \neq j \tag{5.9b}$$

Condition (5.9a) ensures that all actuators are commanded, condition (5.9b) that each actuator is commanded by only one controller. For the sake of simplicity of notation, since now on we assume that $M = m$ and that $I_{ui}^{\#} = i$, $i = 1, \ldots, m$, *i.e.*, that each controller #i only controls the ith input signal. As observed earlier, in general $\mathscr{I}_{x^i} \cap \mathscr{I}_{x^j} \neq \emptyset$, meaning that controller #i may partially share the same feedback information with controller #j, and $\mathscr{I}_{ui} \cap \mathscr{I}_{uj} \neq \emptyset$. This means that controller #i may take into account the effect of control actions that are actually decided by another controller #j, $i \neq j$, $i, j = 1, \ldots, M$, which ensures a certain degree of cooperation.

The designer has the flexibility of choosing the pairs (W_i, Z_i) of decoupling matrices, $i = 1, \ldots, M$. A first guess of the decoupling matrices can be inspired by the intensity of the dynamical interactions existing in the model. The larger (n_i, m_i) the smaller the model mismatch and hence the better the performance of the overall-closed loop system. On the other hand, the larger (n_i, m_i) the larger is the communication and computation efforts of the controllers, and hence the larger the sampling time of the controllers. An example of model decomposition is given later in Section 5.4.1.

5.3.1.2 Decentralized Optimal Control Problems

In order to exploit submodels (5.8) for formulating local finite-horizon optimal control problems that lead to an overall closed-loop stable DMPC system, let the following assumptions be satisfied (these will be relaxed in Theorem 5.5):

Assumption 5.1. *Matrix A in (5.1) is strictly Hurwitz*[1].

Assumption 5.1 restricts the strategy and stability results of DMPC to processes that are open-loop asymptotically stable, leaving to the controller the mere role of optimizing the performance of the closed-loop system.

Assumption 5.2. *Matrix A_i is strictly Hurwitz, $\forall i = 1, \ldots, M$.*

Assumption 5.2 restricts the degrees of freedom in choosing the decentralized models. Note that if $A_i = A$ for all $i = 1, \ldots, M$ is the only choice satisfying Assumption 5.2, then no decentralized MPC can be formulated within this framework. For all $i = 1, \ldots, M$ consider the following infinite-horizon constrained optimal control problems

$$V_i(x(t)) = \min_{\{u_k^i\}_{k=0}^{\infty}} \sum_{k=0}^{\infty} (x_k^i)' W_i' Q W_i x_k^i + (u_k^i)' Z_i' R Z_i u_k^i = \tag{5.10a}$$

$$= \min_{u_0^i} (x_1^i)' P_i x_1^i + x^i(t)' W_i' Q W_i x^i(t) + (u_0^i)' Z_i' R Z_i u_0^i \tag{5.10b}$$

$$\text{s.t. } x_1^i = A_i x^i(t) + B_i u_0^i \tag{5.10c}$$

$$x_0^i = W_i' x(t) = x^i(t) \tag{5.10d}$$

$$u_{\min}^i \leq u_0^i \leq u_{\max}^i \tag{5.10e}$$

$$u_k^i = 0,\ \forall k \geq 1 \tag{5.10f}$$

where $P_i = P_i' \geq 0$ is the solution of the Lyapunov equation

$$A_i' P_i A_i - P_i = -W_i' Q W_i \tag{5.11}$$

that exists by virtue of Assumption 5.2. Problem (5.10) corresponds to a finite-horizon constrained problem with control horizon $N_u = 1$ and linear stable prediction model. Note that only the local state vector $x^i(t)$ is needed to solve Problem (5.10).

[1] While usually a matrix A is called *Hurwitz* if all its eigenvalues have strictly negative real part (continuous-time case), in this paper we say that a matrix A is *Hurwitz* if all the eigenvalues λ_i of A are such that $|\lambda_i| < 1$ (discrete-time case).

At time t, each controller MPC #i measures (or estimates) the state $x^i(t)$ (usually corresponding to local and neighboring states), solves problem (5.10) and obtains the optimizer

$$u_0^{*i} = [u_0^{*i1}, \ldots, u_0^{*ii}, \ldots, u_0^{*im_i}]' \in \mathbb{R}^{m_i} \tag{5.12}$$

In the simplified case $M = m$ and $I_{ui}^{\#} = i$, only the i-th sample of u_0^{*i}

$$u_i(t) = u_0^{*ii} \tag{5.13}$$

will determine the i-th component $u_i(t)$ of the input vector actually commanded to the process at time t. The inputs u_0^{*ij}, $j \neq i$, $j \in \mathscr{I}_{ui}$ to the neighbors are discarded, their only role is to provide a better prediction of the state trajectories x_k^i, and therefore a possibly better performance of the overall closed-loop system.

The collection of the optimal inputs of all the M MPC controllers,

$$u(t) = [u_0^{*11} \; \ldots \; u_0^{*ii} \; \ldots \; u_0^{*mm}]' \tag{5.14}$$

is the actual input commanded to process (5.1). The optimizations (5.10) are repeated at time $t+1$ based on the new states $x^i(t+1) = W_i' x(t+1)$, $\forall i = 1, \ldots, M$, according to the usual receding horizon control paradigm. The degree of coupling between the DMPC controllers is dictated by the choice of the decoupling matrices (W_i, Z_i). Clearly, the larger the number of interactions captured by the submodels, the more complex the formulation (and, in general, the solution) of the optimization problems (5.10) and hence the computations performed by each control agent. Note also that a higher level of information exchange between control agents requires a higher communication overhead. We are assuming here that the submodels are constant at all time steps.

5.3.1.3 Convergence Properties

As mentioned in the introduction, one of the major issues in decentralized RHC is to ensure the stability of the overall closed-loop system. The non-triviality of this issue is due to the fact that the prediction of the state trajectory made by MPC #i about state $x^i(t)$ is in general not correct, because of partial state and input information and of the mismatch between u^{*ij} (desired by controller MPC #i) and u^{*jj} (computed and applied to the process by controller MPC #j).

The following theorem, proved in [1, 2], summarizes the closed-loop convergence results of the proposed DMPC scheme using the cost function $V(x(t)) \triangleq \sum_{i=1}^{M} V_i(W_i' x(t))$ as a Lyapunov function for the overall system.

Theorem 5.3. *Let Assumptions 5.1, 5.2 hold and define P_i as in* (5.11) $\forall i = 1, \ldots, M$. *Define*

$$\begin{aligned} \Delta u^i(t) &\triangleq u(t) - Z_i u_0^{*i}(t), & \Delta x^i(t) &\triangleq (I - W_i W_i')x(t) \\ \Delta A^i &\triangleq (I - W_i W_i')A, & \Delta B^i &\triangleq B - W_i W_i' B Z_i Z_i' \end{aligned} \tag{5.15}$$

Also, let

$$\Delta Y^i(x(t)) \triangleq W_iW_i'(A\Delta x^i(t) + BZ_iZ_i'\Delta u^i(t)) + \Delta A^i x(t) + \Delta B^i u(t) \tag{5.16a}$$

and

$$\Delta S^i(x(t)) \triangleq \big(2(A_iW_i'x(t) + B_iu_0^{*i}(t))' + \Delta Y^i(x(t))'W_i\big)P_iW_i'\Delta Y^i(x(t)) \tag{5.16b}$$

If the condition

$$(i)\ x'\left(\sum_{i=1}^{M} W_iW_i'QW_iW_i'\right)x - \sum_{i=1}^{M}\Delta S^i(x) \geq 0,\ \forall x \in \mathbb{R}^n \tag{5.17a}$$

is satisfied, or the condition

$$(ii)\ x'\left(\sum_{i=1}^{M} W_iW_i'QW_iW_i'\right)x - \alpha x'x - \sum_{i=1}^{M}\Delta S^i(x) + \sum_{i=1}^{M} u_0^{*i}(x)'Z_i'RZ_iu_0^{*i}(x) \geq 0,$$
$$\forall x \in \mathbb{R}^n \tag{5.17b}$$

is satisfied for some scalar $\alpha > 0$*, then the decentralized MPC scheme defined in (5.10)–(5.14) in closed loop with (5.1) is globally asymptotically stable.*

By using the explicit MPC results of [9], each optimizer function $u_0^{*i} : \mathbb{R}^n \mapsto \mathbb{R}^{m_i}$, $i = 1, \ldots, M$, can be expressed as a piecewise affine function of x

$$u_0^{*i}(x) = F_{ij}x + G_{ij} \quad \text{if} \quad H_{ij}x \leq K_{ij}, \quad j = 1, \ldots, n_{ri} \tag{5.18}$$

Hence, both condition (5.17a) and condition (5.17b) are a composition of quadratic and piecewise affine functions, so that *global stability* can be tested through linear matrix inequality relaxations [19] (cf. the approach of [16]).

As $u_{\min} < 0 < u_{\max}$, there exists a ball around the origin $x = 0$ contained in one of the regions, say $\{x \in \mathbb{R}^n :\ H_{i1}x \leq K_{i1}\}$, such that $G_{i1} = 0$. Therefore, around the origin both (5.17a) and (5.17b) become a quadratic form $x'(\sum_{i=1}^{M} E_i)x$ of x, where matrices E_i can be easily derived from (5.15), (5.16) and (5.17). Hence, *local stability* of (5.10)–(5.14) in closed loop with (5.1) can be simply tested by checking the positive semidefiniteness of the square $n \times n$ matrix $\sum_{i=1}^{M} E_i$. Note that, depending on the degree of decentralization, in order to satisfy the sufficient stability criterion one may need to set $Q > 0$ in order to dominate the unmodeled dynamics arising from the terms ΔS^i.

In the absence of input constraints, Assumptions 5.1, 5.2 can be removed to extend the previous DMPC scheme to the case where (A, B) and (A_i, B_i) may not be Hurwitz, although stabilizable.

Theorem 5.4 ([1, 3]). *Let the pairs* (A_i, B_i) *be stabilizable,* $\forall i = 1, \ldots, M$*. Let Problem (5.10) be replaced by*

$$V_i(x(t)) = \min_{\{u_k^i\}_{k=0}^{\infty}} \sum_{k=0}^{\infty} (x_k^i)' W_i' Q W_i x_k^i + (u_k^i)' Z_i' R Z_i u_k^i = \tag{5.19a}$$

$$= \min_{u_0^i} (x_1^i)' P_i x_1^i + x^i(t)' W_i' Q W_i x^i(t) + (u_0^i)' Z_i' R Z_i u_0^i \tag{5.19b}$$

$$\text{s.t. } x_1^i = A_i x^i(t) + B_i u_0^i \tag{5.19c}$$

$$x_0^i = W_i' x(t) = x^i(t) \tag{5.19d}$$

$$u_k^i = K_{LQ_i} x_k^i, \ \forall k \geq 1 \tag{5.19e}$$

where $P_i = P_i' \geq 0$ is the solution of the Riccati equation

$$W_i' Q W_i + K_{LQ_i}' Z_i' R Z_i K_{LQ_i} + (A_i + B_i K_{LQ_i})' P_i (A_i + B_i K_{LQ_i}) = P_i \tag{5.20}$$

and $K_{LQ_i} = -(Z_i' R Z_i + B_i' P_i B_i)^{-1} B_i' P_i A_i$ is the corresponding local LQR feedback. Let $\Delta Y^i(x(t))$ and let $\Delta S^i(x(t))$ be defined as in (5.16)*, in which P_i is defined as in* (5.20).

If condition (5.17a) *is satisfied, or condition* (5.17b) *is satisfied for some scalar $\alpha > 0$, then the decentralized MPC scheme defined in (5.19), (5.14) in closed-loop with (5.1) is globally asymptotically stable.*

So far we assumed that the communication model among neighboring MPC controllers is faultless, so that each MPC agent successfully receives the information about the states of its corresponding submodel. However, one of the main issues in networked control systems is the unreliability of communication channels, which may result in data packet dropout.

A sufficient condition for ensuring convergence of the DMPC closed-loop in the case packets containing measurements are lost for an arbitrary but upper-bounded number N of consecutive time steps was proved in [1, 5]. The underlying operating assumption is that if the actual number of lost packets exceeds the given N, the decentralized controllers are turned off and $u = 0$ is applied persistently, so that a number of packet drops larger than N is not considered. The results shown here are based on formulation (5.10) and rely on the open-loop asymptotic stability Assumptions 5.1 and 5.2. The issue is still non-trivial, as if a set of measures for subsystem i is lost, this would affect not only the trajectory of subsystem i because of the improper control action u^i, but, due to the dynamical coupling, also the trajectories of subsystems $j \in J$, where $J = \{j \mid i \in \mathscr{I}_{xj} \cup \mathscr{I}_{uj}\}$, and thus the closed-loop stability of the overall system may be endangered.

By relying on open-loop stability, setting $u(t) = 0$ is a natural choice for backup input moves when no state measurements are available because of a communication blackout. Different backup options may be considered, such as solving (5.10) by replacing $x^i(t)$ with an estimate obtained through model (5.8) and the available measurements. Of course whether one or the other approach is better strongly depends on the amount of model mismatch introduced by the decentralized modeling.

The next theorem, proved in [1], provides conditions for asymptotic closed-loop stability of decentralized MPC under packet loss, generalizing and unifying the results of [2, 3].

Theorem 5.5. *Let N be a positive integer such that no more than N consecutive steps of channel transmission blackout can occur. Assume $u(t) = 0$ is applied when no packet is received. Let Assumptions 5.1, 5.2 hold and $\forall i = 1,\dots,M$ define P_i as in* (5.11)*, $\Delta u^i(t)$, $\Delta x^i(t)$, ΔA^i, ΔB^i as in* (5.15)*, $\Delta Y^i(x(t))$ as in* (5.16a)*,*

$$\Delta S^i_j(x) \triangleq [2(A_iW_i'x + B_iu_0^{*i}(x))'W_i' + \Delta Y^i(x)'](A^{j-1})'W_iP_iW_i'A^{j-1}\Delta Y^i(x) \quad (5.21)$$

and let

$$\xi_i(x) \triangleq A_iW_i'x + B_iu_0^{*i}(x)$$

If the condition

$$(i)\ \sum_{i=1}^{M} \left(x'W_iW_i'QW_iW_i'x + \xi_i(x)'(P_i - W_i'(A^{j-1})'W_iP_iW_i'A^{j-1}W_i)\xi_i(x) - \Delta S^i_j(x)\right) \geq 0, \quad \forall x \in \mathbb{R}^n,\ \forall j = 1,\dots,N \quad (5.22a)$$

is satisfied, or the condition

$$(ii)\ \sum_{i=1}^{M} \left(x'W_iW_i'QW_iW_i'x + \xi_i(x)'(P_i - W_i'(A^{j-1})'W_iP_iW_i'A^{j-1}W_i)\xi_i(x) + {} -\Delta S^i_j(x) + u_0^{*i}(x)'Z_i'RZ_iu_0^{*i}(x)\right) \alpha x'x \geq 0, \qquad \forall x \in \mathbb{R}^n,\ \forall j = 1,\dots,N \quad (5.22b)$$

is satisfied for some scalar $\alpha > 0$, then the decentralized MPC scheme defined in (5.10)–(5.14) in closed loop with (5.1) is globally asymptotically stable under packet loss.

Note again that around the origin the conditions in (5.22) become a quadratic form to be checked positive semidefinite, so local stability of (5.10)–(5.14) in closed loop with (5.1) under packet loss can be tested for any arbitrary fixed N. Note also that conditions (5.22) are a generalization of (5.17), as for $j = 1$ (no packet drop) matrix $P_i - W_i'(A^{j-1})'W_iP_iW_i'A^{j-1}W_i = P_i - P_i = 0$.

5.3.1.4 Extension to Set-Point Tracking

Consider the following discrete-time linear global process model

$$\begin{cases} z(t+1) = Az(t) + Bv(t) + F_d(t) \\ h(t) = Cz(t) \end{cases} \quad (5.23)$$

where $z \in \mathbb{R}^n$ is the state vector, $v \in \mathbb{R}^m$ is the input vector, $y \in \mathbb{R}^p$ is the output vector, $F_d \in \mathbb{R}^d$ is a vector of measured disturbances. Let A satisfy Assumption 5.1 and assume F_d is constant. The considered set-point tracking problem is that of h tracking a given reference value $r \in \mathbb{R}^p$, despite the presence of F_d. In order to recast the problem as a regulation problem, assume steady-state vectors $z_r \in \mathbb{R}^n$ and $v_r \in \mathbb{R}^m$ exist solving the static problem

$$\begin{cases} (I-A)z_r = Bv_r + F_d \\ r = Cz_r \end{cases} \tag{5.24}$$

and let $x = z - z_r$ and $u = v - v_r$. Input constraints $v_{\min} \leq v \leq v_{\max}$ are mapped into constraints $v_{\min} - v_r \leq u \leq v_{\max} - v_r$[2].

Proposition 5.1. *Under the global coordinate transformation* (5.24)*, the process* (5.23) *under the decentralized MPC law* (5.10)–(5.14) *is such that* $h(t)$ *converges asymptotically to the set-point r, either under the assumption of Theorem 5.3 or, in the presence of packet drops, of Theorem 5.5.*

Note that problem (5.24) is solved in a *centralized* way. Defining local coordinate transformations v_{ir}, z_{ir} based on submodels (5.8) would not lead, in general, to offset-free tracking, due to the mismatch between global and local models. This is a general observation one needs to take into account in decentralized tracking. Note also that both v_r and z_r depend on F_d as well as r, so problem (5.24) should be solved each time the value of F_d or r change and retransmitted to each controller.

5.3.2 DMPC Approach of Jia and Krogh

In [17, 12] the system under control is composed by a number of unconstrained linear discrete-time subsystems with decoupled input signals, described by the equations

$$\begin{bmatrix} x_1(k+1) \\ \vdots \\ x_M(k+1) \end{bmatrix} = \begin{bmatrix} A_{11} & \dots & A_{1M} \\ \vdots & \ddots & \vdots \\ A_{M1} & \dots & A_{MM} \end{bmatrix} \begin{bmatrix} x_1(k) \\ \vdots \\ x_M(k) \end{bmatrix} + \begin{bmatrix} B_1 & & 0 \\ & \ddots & \\ 0 & & B_M \end{bmatrix} \begin{bmatrix} u_1(k) \\ \vdots \\ u_M(k) \end{bmatrix} \tag{5.25}$$

The effect of dynamical coupling between neighboring states is modeled in prediction through a disturbance signal v, for instance the prediction model used by controller $\#j$ is

$$x_j(k+i+i|k) = A_{jj}x_j(k+i|k) + B_j + u_j(k+i|k) + K_j v_j(k+i|k) \tag{5.26}$$

where $K_j = [A_{j1} \ \dots \ A_{j,j-1} \ A_{j,j+1} \ \dots \ A_{jM}]$. The information exchanged between control agents at the end of each sample step is the entire prediction of the local state vector. In particular, controller $\#j$ receives the signal

[2] In case $v_r \notin [v_{\min}, v_{\max}]$, perfect tracking under constraints is not possible, and an alternative is to set

$$\begin{bmatrix} z_r \\ v_r \end{bmatrix} = \arg\min \left\| \begin{bmatrix} I-A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} z_r \\ v_r \end{bmatrix} - \begin{bmatrix} F_d \\ r \end{bmatrix} \right\| \quad \text{s.t.} \quad v_{\min} \leq v_r \leq v_{\max}$$

$$v_j(k+i|k) = \begin{bmatrix} x_1(k+i|k-1) \\ \vdots \\ x_{j-1}(k+i|k-1) \\ x_{j+1}(k+i|k-1) \\ \vdots \\ x_M(k+i|k-1) \end{bmatrix}$$

where i is the prediction time index, from the other MPC controllers at the end of the previous time step $k-1$. The signal $v_j(k+i|k)$ is used by controller $\#j$ at time k to estimate the effect of the neighboring subsystem dynamics in (5.26).

Under certain assumptions of the model matrix A, closed-loop stability is proved by introducing a contractive constraint on the norm of $x_j(k+1|k)$ in each local MPC problem, which the authors prove to be a recursively feasible constraint.

The authors deal with state constraints in [18] by proposing a min-max approach, at the price of a possible conservativeness of the approach.

5.3.3 DMPC Approach of Venkat, Rawlings, and Wright

In [35, 36, 34] the authors propose distributed MPC algorithm based on a process of negotiations among DMPC agents. The adopted prediction model is

$$\begin{cases} x_{ii}(k+1) = A_{ii}x_{ii}(k) + B_{ii}u_i(k) & \text{(local prediction model)} \\ x_{ij}(k+1) = A_{ij}x_{ij}(k) + B_{ij}u_j(k) & \text{(interaction model)} \\ y_i(k) = \sum_{j=1}^{M} C_{ij}x_{ij}(k) \end{cases}$$

The effect of the inputs of subsystem $\#j$ on subsystem $\#i$ is modeled by using an "interaction model". All interaction models are assumed stable, and constraints on inputs are assumed decoupled (*e.g.*, input saturation).

Starting from a multiobjective formulation, the authors distinguish between a "communication-based" control scheme, in which each controller $\#i$ is optimizing his own local performance index Φ_i, and a "cooperation-based" control scheme, in which each controller $\#i$ is optimizing a weighted sum $\sum_{j=1}^{M} \alpha_j \Phi_j$ of all performance indices, $0 \leq \alpha_j \leq 1$. As performance indices depend on the decisions taken by the other controllers, at each time step k a sequence of iterations is taken before computing and implementing the input vector $u(k)$. In particular, within each sampling time k, at every iteration p the previous decisions $u_{j\neq i}^{p-1}$ are broadcast to controller $\#i$, in order to compute the new iterate u_i^p. With the communication-based approach, the authors show that if the sequence of iterations converges, it converges to a Nash equilibrium. With the cooperation-based approach, convergence to the optimal (centralized) control performance is established. In practical situations the process sampling interval may be insufficient for the computation time required for convergence of the iterative algorithm, with a consequent loss of performance. Nonetheless, closed-loop stability is not compromised: as it is achieved even though the convergence of the iterations is not reached. Moreover, all iterations are plantwide feasible, which naturally increases the applicability of the approach including a certain robustness to transmission faults.

5.3.4 DMPC Approach of Dunbar and Murray

In [15] the authors consider the control of a special class of dynamically decoupled continuous-time nonlinear subsystems

$$\dot{x}_i(t) = f_i(x_i(t), u_i(t))$$

where the local states of each model represent a position and a velocity signal

$$x_i(t) = \begin{bmatrix} q_i(t) \\ \dot{q}_i(t) \end{bmatrix}$$

State vectors are only coupled by a global performance objective

$$L(x,u) = \sum_{(i,j)\in\mathscr{E}_0} \omega\|q_i - q_j + d_{ij}\|^2 + \omega\|q_\Sigma - q_d\|^2 + \nu\|\dot{q}\|^2 + \mu\|u\|^2 \qquad (5.27)$$

under local input constraints $u_i(t) \in U$, $\forall i = 1,\ldots,M$, $\forall t \geq 0$. In (5.27) $\mathscr{E}_0$ is the set of pair-wise neighbors, d_{ij} is the desired distance between subsystems i and j, $q_\Sigma = (q_1 + q_2 + q_3)/3$ is the average position of the leading subsystems 1,2,3, and $q_d = (q_1^c + q_2^c + q_3^c)/3$ the corresponding target.

The overall integrated cost (5.27) is decomposed in distributed integrated cost functions

$$L_i(x_i, x_{-i}, u_i) = L_i^x(x_i, x_{-i}) + \gamma\mu\|u_i\|^2 + L^d(i)$$

where $x_{-i} = (x_{j1}, \ldots, x_{jk})$ collects the states of the neighbors of agent subsystem #i, $L_i^x(x_i, x_{-i}) = \sum_{j\in N_i} \frac{\gamma\omega}{2}\|q_i - q_j + d_{ij}\|^2 + \gamma\nu\|\dot{q}_i\|^2$, and

$$L^d(i) = \begin{cases} \gamma\omega\|q_\Sigma - q_d\|^{2/3} & i \in \{1,2,3\} \\ 0 & \text{otherwise} \end{cases}$$

It holds that

$$L(x,u) = \frac{1}{\gamma}\sum_{i=1}^{N} L_i(x_i, x_{-i}, u_i)$$

Before computing DMPC actions, neighboring subsystems broadcast in a synchronous way their states, and each agent transmits and receives an "assumed" control trajectory $\hat{u}_i(\tau;t_k)$. Denoting by $u_i^p(\tau;t_k)$ the control trajectory predicted by controller #i, by $u_i^*(\tau;t_k)$ the optimal predicted control trajectory, by T the prediction horizon, and by $\delta \in (0,T]$ the update interval, the following DMPC performance index is minimized

$$\begin{aligned}
&\min_{u_i^p} \quad J_i(x_i(t_k), x_{-i}(t_k), u_i^p(\cdot;t_k)) \\
&\quad = \min_{u_i^p} \quad \int_{t_k}^{t_k+T} L_i(x_i^p(s;t_k), \hat{x}_{-i}(s;t_k), u_i^p(s;t_k))ds + \gamma \|x_i^p(t_k+T;t_k) - x_i^C\|_{P_i}^2 \\
&\qquad \text{s.t.} \quad \dot{x}_i^p(\tau;t_k) = f_i(x_i^p(\tau;t_k), u_i^p(\tau;t_k)) \\
&\qquad\qquad \dot{\hat{x}}_i^p(\tau;t_k) = f_i(\hat{x}_i^p(\tau;t_k), \hat{u}_i^p(\tau;t_k)) \\
&\qquad\qquad \dot{\hat{x}}_{-i}^p(\tau;t_k) = f_{-i}(\hat{x}_{-i}^p(\tau;t_k), \hat{u}_{-i}^p(\tau;t_k)) \\
&\qquad\qquad u_i^p(\tau;t_k) \in U \\
&\qquad\qquad \|x_i^p(\tau;t_k) - \hat{x}_i(\tau;t_k)\| \leq \delta^2 \kappa \\
&\qquad\qquad x_i^p(t_k+T;t_k) \in \Omega_i(\varepsilon_i)
\end{aligned}$$

The second last constraint is a "compatibility" constraint, enforcing consistency between what agent #i plans to do and what its neighbors believe it plans to do. The last constraint is a terminal constraint.

Under certain technical assumptions, the authors prove that the DMPC problems are feasible at each update step k, and under certain bounds on the update interval δ convergence to a given set is also proved. Note that closed-loop stability is ensured by constraining the state trajectory predicted by each agent to stay close enough to the trajectory predicted at the previous time step that has been broadcasted. The main drawback of the approach is the conservativeness of the compatibility constraint.

5.3.5 DMPC Approach of Keviczy, Borrelli, and Balas

Dynamically decoupled submodels are also considered in [21], where the special nonlinear discrete-time system structure

$$x_{k+1}^i = f^i(x_k^i, u_k^i)$$

is assumed, subject to local input and state constraints $x_k^i \in \mathscr{X}^i$, $u_k^i \in \mathscr{U}^i$, $i = 1, \ldots, M$. Subsystems are coupled by the cost function

$$l(\tilde{x}, \tilde{u}) = \sum_{i=1}^{N_v} l^i(x^i, u^i, \tilde{x}^i, \tilde{u}^i)$$

and by the global constraints

$$g^{i,j}(x^i, u^i, x^j, u^j) \leq 0, \ (i,j) \in \mathscr{A}$$

where $\mathscr{A}$ is a given set. Each local MPC controller is based on the optimization of the following problem

$$\min_{\tilde{U}_t} \sum_{k=0}^{N-1} l(\tilde{x}_{k,t}, \tilde{u}_{k,t}) + l_N(\tilde{x}_{N,t}) \tag{5.28a}$$

$$\text{s.t.} \quad x^i_{k+1,t} = f^i(x^i_{k,t}, u^i_{k,t}) \tag{5.28b}$$

$$x^i_{k,t} \in \mathscr{X}^i, \quad u^i_{k,t} \in \mathscr{U}^i, \; k = 1, \ldots, N-1 \tag{5.28c}$$

$$x^i_{N,t} \in \mathscr{X}^i_f \tag{5.28d}$$

$$x^j_{k+1,t} = f^j(x^j_{k.t}, u^j_{k,t}), (i,j) \in \mathscr{A} \tag{5.28e}$$

$$x^j_{k,t} \in \mathscr{X}^j, \quad u^j_{k,t} \in \mathscr{U}^j, (i,j) \in \mathscr{A} \; k = 1, \ldots, N-1 \tag{5.28f}$$

$$x^j_{N,t} \in \mathscr{X}^j_f, (i,j) \in \mathscr{A} \tag{5.28g}$$

$$g^{i,j}(x^i_{k,t}, u^i_{k,t}, x^j_{k,t}, u^j_{k,t}) \leq 0, (i,j) \in \mathscr{A} \; k = 1, \ldots, N-1 \tag{5.28h}$$

$$x^i_{0,t} = x^i_t, \tilde{x}^i_{0,t} = \tilde{x}^i_t \tag{5.28i}$$

where (5.28b)–(5.28d) are the local model and constraints of the agent, (5.28e)–(5.28g) are the model and constraints of the neighbors, and (5.28h) represent interaction constraints of agent #i with its own neighbors.

The information exchanged among the local MPC agents are the neighbors' current states, terminal regions, and local models and constraints. As in (5.13), only the optimal input $u^i_{0,t}$ computed by controller #i is applied; the remaining inputs $u^j_{k,t}$ are completely discarded, as they are only used to enhance the prediction.

Stability is analyzed for the problem without coupling constraints (5.28h), under the assumption that the following inequality holds

$$\sum_{k=1}^{N-1} 2\|Q(x^{j,j}_{k,t} - x^{j,i}_{k,t})\|_p + \|R(u^{j,j}_{k,t} - u^{j,i}_{k,t})\|_p \leq \|Qx^i_t\|_p + \|Qx^j_t\|_p + \\ \|Q(x^i_t - x^j_t)\|_p + \|Ru^{i,i}_{0,t}\|_p + \|Ru^{j,i}_{0,t}\|_p$$

where $\|Qx\|_2 \triangleq x'Qx$, and $\|Qx\|_1$, $\|Qx\|_\infty$ are the standard q and ∞ norm, respectively.

5.3.6 DMPC Approach of Mercangöz and Doyle

The distributed MPC *and* estimation problems are considered in [25] for square plants (the number of inputs equals the number of outputs) perturbed by noise, whose local prediction models are

$$\begin{cases} x_i(k+1) = A_i x_i(k) + B_i u_i(k) + \sum_{j=1}^{M} B_j u_j(k) + w_i(k) \\ \quad y_i(k) = C_i x_i(k) + v_i(k) \end{cases} \tag{5.29}$$

A distributed Kalman filter based on the local submodels (5.29) is used for state estimation. The DMPC approach is similar to Venkat et al.'s "communication-based" approach, although only first moves $u_j(k)$ are transmitted and assumed frozen in prediction, instead of the entire optimal sequences. Only constraints on local inputs are handled by the approach. Although general stability and convergence results are not proved in [25], experimental results on a four-tank system are reported to show the effectiveness of the approach.

5.3.7 DMPC Approach of Magni and Scattolini

Another interesting approach to decentralized MPC for nonlinear systems has been formulated in [23]. The problem of regulating a nonlinear system affected by disturbances to the origin is considered under some technical assumptions of regularity of the dynamics and of boundedness of the disturbances. Closed-loop stability is ensured by the inclusion in the optimization problem of a contractive constraint. The considered class of functions and the absence of information exchange between controllers leads to some conservativeness of the approach.

5.4 Example of Decentralized Temperature Control in a Railcar

5.4.1 Example Description

In this section we test the DMPC approach of Alessio, Barcelli, and Bemporad for decentralized control of the temperature in different passenger areas in a railcar [5]. The system is schematized in Figure 5.2. Each passenger area has its own heater and air conditioner but its thermal dynamics interacts with surrounding areas (neighboring passenger areas, external environment, antechambers) directly or through windows, walls and doors. Passenger areas are composed by a table and the associated four seats. Temperature sensors are located in each four-seat area, in each antechamber, and along the corridor. The goal of the controller is to adjust each passenger area to its own temperature set-point to maximize passenger comfort. Temperature sensors may be wired or wireless, in the latter case we assume that information packets may be dropped, because of very low power transmission, simplified transmission protocols, and no acknowledgement and retransmission and because of time-varying communication disturbances due for example to passengers' electronic equipment.

Let $2N$ be the number of four-seat areas ($N = 8$ in Figure 5.2), N the number of corridor partitions, and 2 the number of antechambers. Under the assumption of perfectly mixed fluids in each jth volume, $j = 1,\dots,n$ where $n = 3N + 2$, the heat transmission equations by conduction lead to the linear model

$$\frac{dT_j(\tau)}{d\tau} = \sum_{i=0}^{n} Q_{ij}(\tau) + Q_{uj},\ Q_{ij}(\tau) = \frac{S_{ij}K_{ij}(T_i(\tau)-T_j(\tau))}{C_jL_{ij}},\ j = 1,\dots,n \quad (5.30)$$

where $T_j(\tau)$ is the temperature of volume $\#j$ at time $\tau \in \mathbb{R}$, $T_0(\tau)$ is the ambient temperature outside the railcar, $Q_{ij}(\tau)$ is heat flow due to the temperature difference $T_i(\tau) - T_j(\tau)$ with the neighboring volume $\#i$, S_{ij} is the contact surface area, Q_{uj} is the heat flow of heater $\#j$, K_{ij} is the thermal coefficient that depends on the materials, $C_j = K_c^j V_j$ is the (material dependent) heat capacity coefficient K_c^j times the fluid

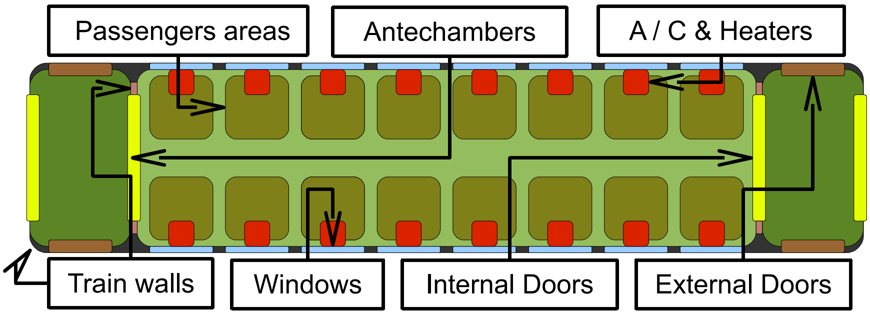


Fig. 5.2 Physical structure of the railcar

volume V_j, and L_{ij} is the thickness of the separator between the two volumes #i and #j. We assume that $Q_{ij}(\tau) = 0$ for all volumes i, j that are not adjacent, $\forall \tau \in \mathbb{R}$. Hence, the process can be modeled as a linear time-invariant continuous-time system with state vector $z \in \mathbb{R}^{3N+2}$ and input vector $v \in \mathbb{R}^{2N}$

$$\begin{cases} \dot{z}(\tau) = A_c z(\tau) + B_c v(\tau) + F T_0(\tau) \\ h(\tau) = C z(\tau) \end{cases} \tag{5.31}$$

where $F \in \mathbb{R}^n$ is a constant matrix, $T_0(\tau)$ is treated as a piecewise constant measured disturbance, and $C \in \mathbb{R}^{p \times n}$ is such that $h \in \mathbb{R}^p$ contains the components of z corresponding to the temperatures of the passenger seat areas, $p = 2N$. Since we assume that the thermal dynamics are relatively slow compared to the sampling time T_s of the decentralized controller we are going to synthesize, we use first-order Euler approximation to discretize dynamics (5.31) without introducing excessive errors:

$$\begin{cases} z(t+1) = A z(t) + B v(t) + F_d T_0(t) \\ \quad h(t) = C z(t) \end{cases} \tag{5.32}$$

where $A = I + A_c T_s$, $B = B_c T_s$, and $F_d = F T_s$. We assume that A is asymptotically stable, as an inheritance of the asymptotic stability of matrix A_c.

In order to track generic temperature references $r(t)$, we adopt the coordinate shift defined by (5.24). The next step is to decentralize the resulting global model. The particular topology of the railcar suggests a decomposition of model (5.1) as the cascade of four-seat areas. There are two kinds of four-seat areas, namely (*i*) the ones next to the antechambers, and (*ii*) the remaining ones. Besides interacting with the external environment, the areas of type (*i*) interact with another four-seat-area, an antechamber, and a section of the corridor, while the areas of type (*ii*) only with the four-seat areas at both sides and the corresponding section of the corridor. Note that the decentralized models overlap, as they share common states and inputs. The decoupling matrices Z_i are chosen so that in each subsystem only the first component of the computed optimal input vector is actually applied to the process.

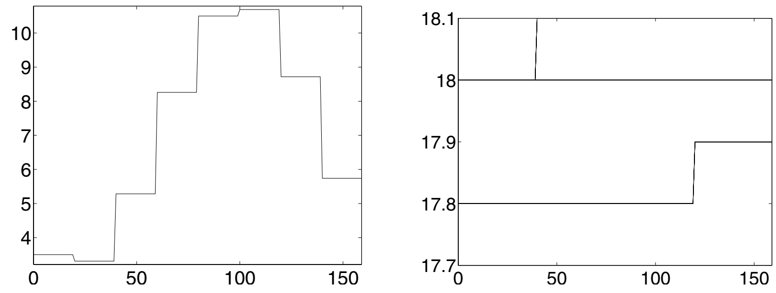


Fig. 5.3 Exogenous signals used in the reported simulations

As a result, each submodel has 5 states and 2 or 3 inputs, depending whether it describes a seat area of type (*i*) or (*ii*), which is considerably simpler than the centralized model (5.1) with 26 states and 16 inputs.

We apply the DMPC approach (5.10) with

$$Q = 2\begin{bmatrix} 10^2 I_{16} & 0 \\ 0 & I_{10} \end{bmatrix},\ R = 10^5 I_{16}, v_{\min} = -0.03\ \text{W},\ v_{\max} = 0.03\ \text{W},\ T_s = 9\ \text{min} \tag{5.33}$$

where $v_{\min}$ is the lower bound on the heat flow subtracted by the air-conditioners, and $v_{\max}$ is the maximum heating power of the heaters (with a slight abuse of notation we denoted by $v_{\min}$, $v_{\max}$ the entries of the corresponding lower and upper bound vectors of $\mathbb{R}^{16}$). Note that the first sixteen diagonal elements of matrix Q correspond to the temperatures of the four-seat areas. It is easy to check that with the parameters in (5.33) condition (5.17a) for local stability is satisfied. For comparison, a centralized MPC approach (5.3) with the same weights, horizon, and sampling time as in (5.33) is also designed. The associated QP problem has 16 optimization variables and 32 constraints, while the complexity of each DMPC controller is either 2 (or 3) variables and 4 (or 6) constraints. The DMPC approach is in fact largely scalable: for longer railcars the complexity of the DMPC controllers remains the same, while the complexity of the centralized MPC problem grows with the increased model size. Note also that, even if a centralized computation is taken, the DMPC approach can be immediately parallelized.

5.4.2 Simulation Results

We investigate different simulation outcomes depending on four ingredients: *i*) type of controller (centralized/decentralized), *ii*) packet-loss probability, *iii*) change in reference values, *iv*) changes of external temperature (acting as a measured disturbance). Figure 5.3 shows the external temperature and reference scenarios used in all simulations.

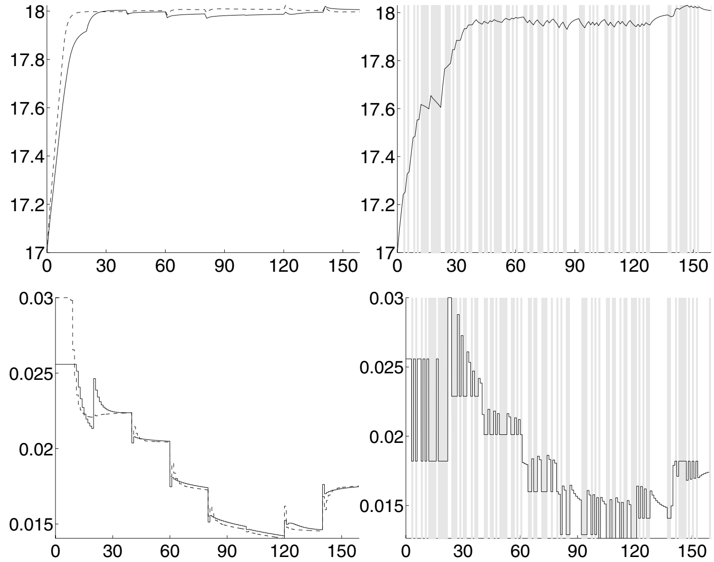


Fig. 5.4 Comparison between centralized MPC (dashed lines) and decentralized MPC (continuous lines): output h_1 (upper plots) and input v_1 (lower plots). Gray areas denote packet drop intervals

In order to compare closed-loop performances in different simulation scenarios, the following performance index

$$J = \sum_{t=1}^{N_{\rm sim}} (z(t) - r(t))' Q (z(t) - r(t)) + (v(t) - v_r)' R (v(t) - v_r) \qquad (5.34)$$

is defined, where $N_{sim} = 160$ (one day) is the total number of simulation steps.

The initial condition is 17°C for all seat-area temperatures, except for the antechamber, which is 15°C. Note that the steady-state value of antechamber temperatures is not relevant for the posed control goals. The closed-loop trajectories of centralized MPC feedback vs. decentralized MPC with no packet-loss are shown in Figure 5.4.2 (we only show the first state and input for clarity). In both cases the temperatures of the four-seat areas converge to the set-point asymptotically. Figure 5.4.2 shows the temperature vector $h(t)$ tracking the time-varying reference $r(t)$ in the absence of packet-loss, where the coordinate transformation (5.24) is repeated after each set-point and external temperature change.

To simulate packet loss, we assume that the probability of losing a packet depends on the state of a Markov chain with N states (see Figure 5.6). We parameterize with the probability parameter p, $0 \leq p \leq 1$ the probabilities associated with the Markov chain: the Markov chain is in the jth state if $j-1$ consecutive packets have been lost. The probability of losing a further packet is $1-p$ in every state of the chain, except for the $(N+1)$th state where no packet can be lost any more.

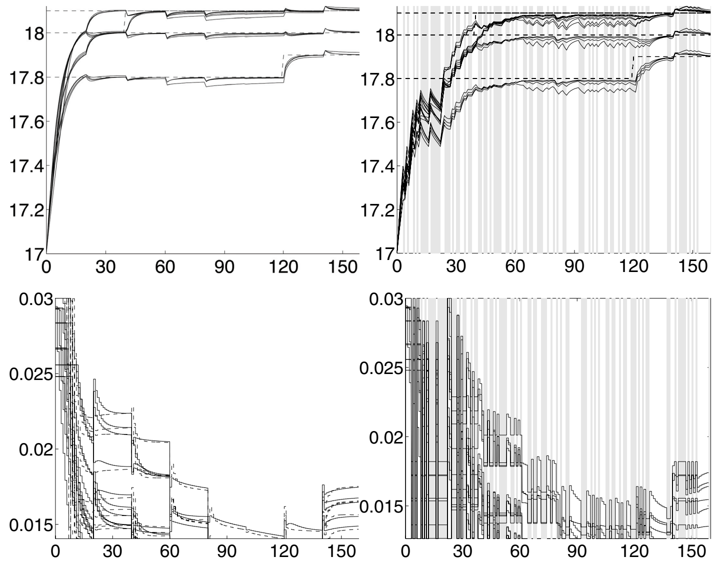


Fig. 5.5 Decentralized MPC results. Upper plots: output variables h (continuous lines) and references r (dashed lines). Lower plots: command inputs v. Gray areas denote packet drop intervals

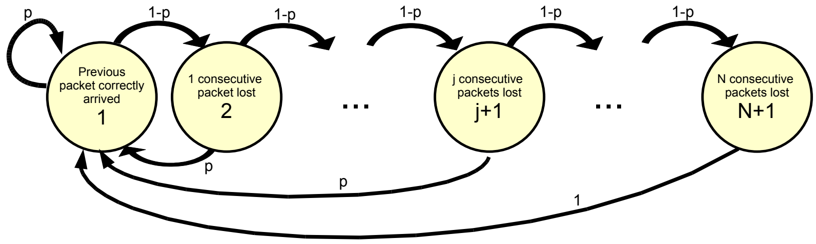


Fig. 5.6 Markov chain model of packet-loss probability

Let π be the stationary probability vector of the Markov chain of Figure 5.6, obtained through the one-step probability matrix

$$P = \begin{bmatrix} p & 1-p & 0 & \cdots & 0 \\ p & 0 & 1-p & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p & 0 & 0 & \cdots & 1-p \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

by solving

$$\begin{cases} \pi' = \pi' P \\ \sum_{i=1}^{N} \pi_i = 1 \end{cases}$$

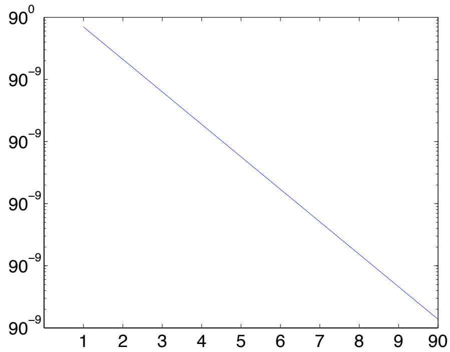


Fig. 5.7 Markov chain packet-loss probability with $N = 10$ and $p = 0.7$

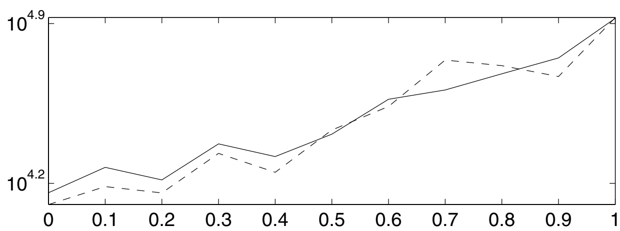


Fig. 5.8 Performance indices of Centralized MPC (dashed line) and Decentralized MPC (solid line)

Recalling the meaning of the Markov chain nodes, the steady-state probability π_i of the i^{th} state is the probability of loosing consecutively exactly $i-1$ packets. The packet-loss discrete probability is shown in Figure 5.7 when $N = 10$ maximum consecutive packet-losses are possible and $p = 0.7$.

Figure 5.7 highlights the exponential decrease of the stationary probabilities as a function of consecutive packets lost. Such a probability model is confirmed by the experimental results on relative frequencies of packet failure burst length observed in [38]. Note that our model assumes that the probability of losing a packet is null after N packets, hence satisfying the assumption of an upper-bound on the number of consecutive drops (as mentioned earlier, we can assume for instance that if $k > N$ consecutive packets are lost, the control loops are shut down). The simulation results obtained with $p = 0.5$ are shown in Figure 5.4.2 and Figure 5.4.2.

In case of packet loss, we also compare the performance of centralized vs. decentralized MPC. Note that in case packet loss occurs also on the communication channel between the point computing the coordinate shift and the decentralized controllers, the last received coordinate shift is kept. The stability condition (5.22a) of Theorem 5.5 was tested and proved satisfied for values of j up to 160.

Figure 5.8 shows that the performance index J defined in Eq. (5.34) increases as the packet-loss probability grows, implying performance to deteriorate due to the conservativeness of the backup control action $u = 0$ (that is, $v = v_r$). The results of Figure 5.8 are averaged over 10 simulations per probability sample. As a general consideration, centralized MPC dominates over the decentralized, although for certain values of p the average performance of decentralized MPC is slightly better, probably due to the particular packet loss sequences that have realized. However, the loss of performance due to decentralization, with regard to the present example, is largely negligible.

The simulations were run on a MacBook Air 1.86 GHz running Matlab R2008a under OS X 10.5.6 and the Hybrid Toolbox for Matlab [7]. The average CPU time for solving the centralized QP problem associated with (5.3) is 6.0ms (11.9ms in the worst case). For the decentralized case, the average CPU time for solving the QP problem associated with (5.10) is 3.3ms (7.4ms in the worst case). Although the decrease of CPU time is only a few milliseconds, we remark that for increasing N the complexity of DMPC remains constant, while the complexity of centralized MPC would grow with N. To quantify this aspect consider that, if one thinks to the explicit form of the MPC controllers [9], the number of regions of the centralized MPC is upper bounded by 3^{16}, while in decentralized case by 3^2 for submodels with two inputs and by 3^3 for submodels with three inputs.

Note that the reference vectors v_r, z_r are computed globally in all simulations. In this example the complexity of such a static calculation is negligible with respect to solving the QP problems. Moreover, the communication burden is also negligible, as new reference vectors are transmitted individually to each MPC agent only when set-point and disturbances change.

5.5 Hierarchical MPC

5.5.1 Problem Description

In this section we provide some novel ideas on how to possibly couple a DMPC layer with a higher centralized (hybrid) MPC layer in the hierarchical setting of Figure 5.1. The idea is to design a centralized hybrid MPC controller to achieve global coordination, namely to enforce global constraints (linear, logical, mixed linear & logical) and to optimize a global objective (such as an economically-driven objective). To achieve the goal, we need an abstract (hybrid) model of the underlying closed-loop dynamics, without resorting to a global dynamical model of the process and to the need of full state feedback.

Under the assumption that the higher MPC layer runs in real-time at a lower sampling frequency than the underlying DMPC layer, a sensible choice is to use the static global model

$$y(k+1) = G(1)u(k)$$

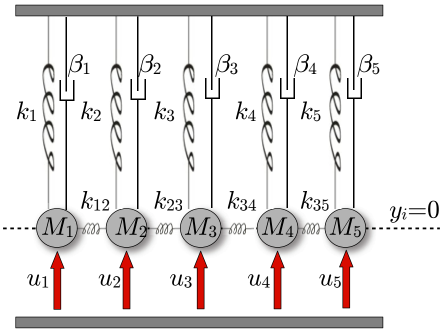


Fig. 5.9 Hierarchical and decentralized MPC of system composed by five dynamically-coupled masses moving vertically

as a centralized abstract model, where $G(1)$ is the DC-gain of (5.1), and k represents the sampling step of the higher-level MPC controller.

Then, the centralized higher-level MPC controller solves the following problem

$$\begin{aligned} \min_{u(k)} \; & f(y(k+1) - r_d(k), u(k)) \\ \text{s.t. } & g(u(k), y(k), r_d(k)) \leq 0 \end{aligned}$$

and defines

$$r(k) = G(1)u(k)$$

as the current setpoints for the DMPC layer. Extensions of these ideas, including quantitative ways of choosing a suitable sampling time for the higher control layer, have been formulated very recently in [6].

5.5.2 Illustrative Example

Consider the system depicted in Figure 5.9, composed by five dynamically-coupled masses moving vertically. The dynamics of each mass #i is described by the dynamics

$$M_i \ddot{y}_i = u_i - \beta_i \dot{y}_i - k_i y_i - \underbrace{k_{ij}(y_i - y_j)}_{j=i-1,i+1} \tag{5.35}$$

where $M_i = 5$ [kg], $\beta_i = 0.1$ [kg/s], $k_i = 1$ [kg/s^2], $k_{ij} = 0.5$ [kg/s^2]. For local prediction purposes, each DMPC controller #i neglects the velocities of the neighboring masses, $\dot{y}_{i-1} = \dot{y}_{i+1} = 0$, and therefore only considers $y_i, \dot{y}_i, y_{i-1}, y_{i+1}$ as local states.

After discretizing the dynamics (5.35) with sampling time $T_L = 0.25$ [s], each DMPC agent solves the following on-line optimal control problem

$$\min_{u_i(k),\ldots,u_i(k+N_u-1)} \sum_{j=0}^{N_y-1} (y_i(k+j)-r_i(k))^2 + 0.1 \sum_{j=0}^{N_u-1} (u_i(k+j)-u_i(k+j-1))^2 + 10^4\varepsilon^2$$

$$\begin{array}{lll} \text{s.t. } & u^i_{\min} \le u_i(k+j) \le u^i_{\max}(k) & j=0,\ldots,N_u-1 \\ & y^i_{\min} - \varepsilon \le y_i(k+j) \le y^i_{\max} + \varepsilon & j=0,\ldots,N \\ & u_i(k+j) = u_i(k+N_u-1) & j=N_u,\ldots,N_y-1 \end{array} \tag{5.36}$$

where $u^i_{\min} = y^i_{\min} = 0$ [m], $y^i_{\max} = 2$ [m], $N_y = 20$ is the prediction horizon, $N_u = 4$ is the control horizon, ε is a slack variable used to soften output constraints to prevent the possible infeasibility of the quadratic program associated with (5.36). The input bounds $u^i_{\max}(k)$ are decided at multiple of the higher-level sampling time $T_H = 3$ [s] by a centralized hybrid MPC, together with the local setpoint vector $r(k)$. The hybrid MPC controller [8, 33] is designed to enforce the following constraints:

(i) at most K_u inputs can be over a certain threshold $u_{\text{lim}} = 0.7$ [m], $u_i(k) \ge u_{\text{lim}}$;
(ii) set-point changes are bounded by a quantity Δ_r=0.5 [m]

$$|G(1)u(k) - y_i(k)| \le \Delta_r \tag{5.37}$$

The logical "at most" constraint (i) is enforced by defining auxiliary binary inputs $u_{\ell i}(k) \in \{0,1\}$, $i = 1,\ldots,5$, and by setting

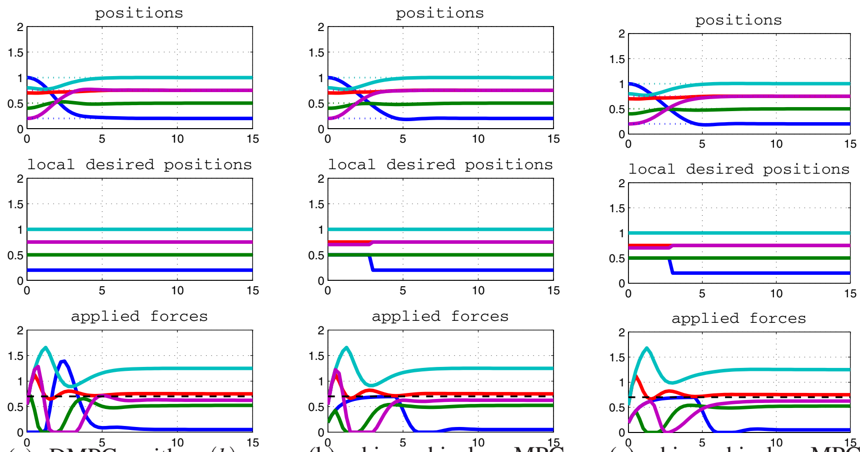
$$u^i_{\max}(k) = \begin{cases} y^i_{\max} & \text{if } u_{\ell i}(k) = 1 \\ u_{\text{lim}} & \text{if } u_{\ell i}(k) = 0 \end{cases} \tag{5.38}$$

By letting $r_d(k) = [0.2\ 0.5\ 0.75\ 1\ 0.75]'$ [m] be the vector of desired vertical positions of the masses, every $T_H/T_L = 12$ steps the hybrid MPC controller solves the following problem

$$\min_{u(k)} \|G(1)u(k) - r_d(k)\|^2 + \sum_{i=1}^{5} (u_{\ell i} - 1)^2$$

subject to the linear constraints defined by (5.37) and the mixed-integer reformulation of constraint (5.38) [8] to determine the reference vector $r(k) = G(1)u(k)$ for the DMPC layer, and the input upper limit $u^i_{\max}(k)$, which is set either to $y^i_{\max}$ or to u_{lim}, depending on the logical constraint.

We simulate the hierarchical control system from initial positions $y_1(0) = 1$, $y_2(0) = 0.4$, $y_3(0) = 0.7$, $y_4(0) = 0.8$, $y_5(0) = 0.2$ and null velocities. The closed-loop results are reported in Figure 5.10. Note that in the absence of the higher-level hybrid MPC controller the red, purple, cyan, and blue input force signals ($u_i(k)$) overpass the limit threshold $u_{\text{lim}}(k)$ (Figure 5.10(a)). When the hybrid MPC controller is used with $K_u = 3$, we obtain the plots depicted in Figure 5.10(b), where only the red, purple, and cyan input signals are set greater than $u_{\text{lim}}(k)$. For $K_u = 2$, we obtain the plots depicted in Figure 5.10(c), where only the red and purple input signals get above $u_{\text{lim}}(k)$.



(a) DMPC with $r(k) = r_d(k)$ (no hybrid MPC)

(b) hierarchical MPC, $K_u = 3$

(c) hierarchical MPC, $K_u = 2$

Fig. 5.10 Hierarchical and decentralized MPC results for the five-mass system. The limit $u_{\lim}$ is shown as a dashed black line

Both the linear DMPC and the hybrid MPC controllers were implemented in Simulink using the Hybrid Toolbox for MATLAB [7].

5.6 Conclusions

In this chapter we have surveyed different approaches to the problem of controlling a distributed process through the cooperation of multiple decentralized model predictive controllers. Each controller is based on a submodel of the overall process, and different submodels may share common states and inputs, to possibly decrease modeling errors in case of dynamical coupling, and to increase the level of cooperativeness of the controllers. The DMPC approach is suitable for control of large-scale systems subject to constraints: the possible loss of global optimal performance is compensated by the gain in controller scalability, reconfigurability, and maintenance. Although a few contributions have been given in the last few years, the DMPC theory is not yet mature and homogenous. In this chapter we have tried to highlight similarities and differences among the various approaches that have been proposed, a little step towards the consolidation of a general theoretical framework for DMPC design.

Open research topics in DMPC include: systematic ways to decompose the model into local submodels, when this is not obvious from the physics of the process, determining the optimal model decomposition (*i.e.*, the best achievable closed-loop performance) for a given channel capacity and computer power available to the control agents; better awareness of DMPC algorithms of the communication efforts, especially when operating over wireless sensor networks (for instance, to save battery energy and hence increase the device life span); stochastic DMPC formulations to take into account imperfect communication in a less conservative way than robust approaches; output-feedback DMPC by using suitable complementary decentralized estimation schemes; hierarchical MPC schemes, for instance combining centralized hybrid MPC and decentralized linear MPC; design better distributed MPC algorithms by taking into account the progress in distributed optimization approaches (*e.g.*, to handle coupled input and state constraints), as described in Chapter 3 of this book.

Acknowledgment

The authors acknowledge the help of Bruce Krogh, Dong Jia, Francesco Borrelli, Richard Murray, William Dunbar, Aswin Venkat, James Rawlings, Mehmet Mercangoz, and Francis Doyle III for sharing copies of their presentations related to their papers on DMPC. Mikael Johansson is also acknowledged for pointing out the packet loss results of [38].

References

1. Alessio, A., Barcelli, D., Bemporad, A.: Decentralized model predictive control of dynamically-coupled linear systems. Technical report. Available upon request from the authors
2. Alessio, A., Bemporad, A.: Decentralized model predictive control of constrained linear systems. In: Proc. European Control Conf., Kos, Greece, pp. 2813–2818 (2007)
3. Alessio, A., Bemporad, A.: Stability conditions for decentralized model predictive control under packet dropout. In: Proc. American Contr. Conf., Seattle, WA, pp. 3577–3582 (2008)
4. Bamieh, B., Paganini, F., Dahleh, M.A.: Distributed control of spatially invariant systems. IEEE Trans. Automatic Control 47(7), 1091–1107 (2002)
5. Barcelli, D., Bemporad, A.: Decentralized model predictive control of dynamically-coupled linear systems: Tracking under packet loss. In: 1st IFAC Workshop on Estimation and Control of Networked Systems, Venice, Italy, pp. 204–209 (2009)
6. Barcelli, D., Bemporad, A., Ripaccioli, G.: Hierarchical multi-rate control design for constrained linear systems. In: Proc. 49th IEEE Conf. on Decision and Control, Atlanta, Georgia USA (to appear, 2010)
7. Bemporad, A.: Hybrid Toolbox – User's Guide (December 2003), `http://www.dii.unisi.it/hybrid/toolbox`
8. Bemporad, A., Morari, M.: Control of systems integrating logic, dynamics, and constraints. Automatica 35(3), 407–427 (1999)
9. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N.: The explicit linear quadratic regulator for constrained systems. Automatica 38(1), 3–20 (2002)
10. Borrelli, F., Keviczky, T., Fregene, K., Balas, G.J.: Decentralized receding horizon control of cooperative vechicle formations. In: Proc. 44th IEEE Conf. on Decision and Control and European Control Conf., Sevilla, Spain, pp. 3955–3960 (2005)
11. Callier, F.M., Chan, W.S., Desoer, C.A.: Input-output stability theory of interconnected systems using decomposition techniques. IEEE Trans. Circuits and Systems 23(12), 714–729 (1976)

12. Camponogara, E., Jia, D., Krogh, B.H., Talukdar, S.: Distributed model predictive control. IEEE Control Systems Magazine, 44–52 (February 2002)
13. Damoiseaux, A., Jokic, A., Lazar, M., van den Bosch, P.P.J., Hiskens, I.A., Alessio, A., Bemporad, A.: Assessment of decentralized model predictive control techniques for power networks. In: 16th Power Systems Computation Conference, Glasgow, Scotland (2008)
14. D'Andrea, R.: A linear matrix inequality approach to decentralized control of distributed parameter systems. In: Proc. American Contr. Conf., pp. 1350–1354 (1998)
15. Dunbar, W.B., Murray, R.M.: Distributed receding horizon control with application to multi-vehicle formation stabilization. Automatica 42(4), 549–558 (2006)
16. Grieder, P., Morari, M.: Complexity reduction of receding horizon control. In: Proc. 42nd IEEE Conf. on Decision and Control, Maui, Hawaii, USA, pp. 3179–3184 (2003)
17. Jia, D., Krogh, B.: Distributed model predictive control. In: Proc. American Contr. Conf., Arlington, VA, pp. 2767–2772 (2001)
18. Jia, D., Krogh, B.: Min-max feedback model predictive control for distributed control with communication. In: Proc. American Contr. Conf., pp. 4507–4512. Anchorage, Alaska (2002)
19. Johannson, M., Rantzer, A.: Computation of piece-wise quadratic Lyapunov functions for hybrid systems. IEEE Trans. Automatic Control 43(4), 555–559 (1998)
20. Johansson, B., Keviczky, T., Johansson, M., Johansson, K.H.: Subgradient methods and consensus algorithms for solving convex optimization problems. In: Proc. 47th IEEE Conf. on Decision and Control, Cancun, Mexico, pp. 4185–4190 (2008)
21. Keviczky, T., Borrelli, F., Balas, G.J.: Decentralized receding horizon control for large scale dynamically decoupled systems. Automatica 42(12), 2105–2115 (2006)
22. Li, W., Cassandras, C.G.: Stability properties of a receding horizon contoller for cooperating UAVs. In: Proc. 43th IEEE Conf. on Decision and Control, Paradise Island, Bahamas, pp. 2905–2910 (2004)
23. Magni, L., Scattolini, R.: Stabilizing decentralized model predictive control of nonlinear systems. Automatica 42(7), 1231–1236 (2006)
24. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: Stability and optimality. Automatica 36(6), 789–814 (2000)
25. Mercangöz, M., Doyle III., F.J.: Distributed model predictive control of an experimental four-tank system. Journal of Process Control 17(3), 297–308 (2007)
26. Michel, A.N.: Stability analysis of interconnected systems. SIAM J. Contr. 12, 554–579 (1974)
27. Michel, A.N., Rasmussen, R.D.: Stability of stochastic composite systems. IEEE Trans. Automatic Control 21, 89–94 (1976)
28. Richards, A., How, J.P.: Decentralized model predictive control of cooperating UAVs. In: Proc. 43rd IEEE Conf. on Decision and Control, Paradise Island, Bahamas (2004)
29. Rotkowitz, M., Lall, S.: A characterization of convex problems in decentralized control. IEEE Trans. Automatic Control 51(2), 1984–1996 (2006)
30. Samar, S., Boyd, S., Gorinevsky, D.: Distributed estimation via dual decomposition. In: Proc. European Control Conf., Kos, Greece, pp. 1511–1516 (2007)
31. Sandell, N.R., Varaiya, P., Athans, M., Safonov, M.G.: Survey of decentralized control methods for large scale systems. IEEE Trans. Automatic Control 23(2), 108–128 (1978)
32. Scattolini, R.: Architectures for distributed and hierarchical model predictive control – a review. Journal of Process Control 19, 723–731 (2009)
33. Torrisi, F.D., Bemporad, A.: HYSDEL — A tool for generating computational hybrid models. IEEE Trans. Contr. Systems Technology 12(2), 235–249 (2004)

34. Venkat, A.N., Hiskens, I.A., Rawlings, J.B., Wright, S.J.: Distributed MPC strategies with application to power system automatic generation control. IEEE Transactions on Control Systems Technology 16(6), 1192–1206 (2008)
35. Venkat, A.N., Rawlings, J.B., Wright, J.S.: Stability and optimality of distributed model predictive control. In: Proc. 44th IEEE Conf. on Decision and Control and European Control Conf., Seville, Spain (2005)
36. Venkat, A.N., Rawlings, J.B., Wright, J.S.: Implementable distributed model predictive control with guaranteed performance properties. In: Proc. American Contr. Conf., Minneapolis, MN, pp. 613–618 (2006)
37. Wang, S., Davison, E.J.: On the stabilization of decentralized control systems. IEEE Trans. Automatic Control 18(5), 473–478 (1973)
38. Willig, A., Mitschke, R.: Results of bit error measurements with sensor nodes and casuistic consequences for design of energy-efficient error control schemes. In: Römer, K., Karl, H., Mattern, F. (eds.) EWSN 2006. LNCS, vol. 3868, pp. 310–325. Springer, Heidelberg (2006)