Distributed Model Predictive Control: A Tutorial Review

Panagiotis D. Christofides, Riccardo Scattolini, David Muñoz de la Peña and Jinfeng Liu

Abstract— **In this paper, we provide a tutorial review of recent results in the design of distributed model predictive control systems. Our goal is to not only conceptually review the results in this area but also to provide enough algorithmic details so that the advantages and disadvantages of the various approaches can be become quite clear. In this sense, our hope is that this paper would complement a series of recent review papers and catalyze future research in this rapidly-evolving area.**

I. INTRODUCTION

Continuously faced with the requirements of safety, environmental sustainability and profitability, chemical process operation has been extensively relying on automated control systems. This realization has motivated extensive research, over the last fifty years, on the development of advanced operation and control strategies to achieve safe, environmentally-friendly and economically-optimal plant operation by regulating process variables at appropriate values. Classical process control systems, like proportional-integral-derivative (PID) control, utilize measurements of a single process output variable (e.g., temperature, pressure, level, or product species concentration) to compute the control action needed to be implemented by a control actuator so that this output variable can be regulated at a desired set-point value. PID controllers have a long history of success in the context of chemical process control and will undoubtedly continue to play an important role in the process industries. In addition to relative ease of implementation, maintenance and organization of a process control system that uses multiple single-loop PID controllers, an additional advantage is the inherent fault-tolerance of such a decentralized control architecture since failure (or poor tuning) of one PID controller or of a control loop does not necessarily imply failure of the entire process control system. On the other hand, decentralized control systems, like the ones based on multiple single-loop PID controllers, do not account for the occurrence of interactions between plant components (subsystems) and control loops, and this may severely limit the best achievable closed-loop performance. Motivated by these issues, a vast array of tools have been developed (most of those included in process control textbooks, e.g., [119], [131], [138]) to quantify these input/output interactions, optimally select the input/output pairs and tune the PID controllers.

While there are very powerful methods for quantifying decentralized control loop interactions and optimizing their performance, the lack of directly accounting for multivariable interactions has certainly been one of the main factors that motivated early on the development of model-based centralized control architectures, ranging from linear pole-placement and linear optimal control to linear model predictive control (MPC). In the centralized approach to control system design, a single multivariable control system is designed that computes in each sampling time the control actions of all the control actuators accounting explicitly for multivariable input/output interactions as captured by the process model. While the early centralized control efforts considered mainly linear process models as the basis for controller design, over the last twenty-five years, significant progress has been made on the direct use of nonlinear models for control system design. A series of papers in previous CPC meetings (e.g., [73], [4], [95], [75], [159]) and books (e.g., [61], [25], [128]) have detailed the developments in nonlinear process control ranging from geometric control to Lyapunov-based control to nonlinear model predictive control.

Independently of the type of control system architecture and type of control algorithm utilized, a common characteristic of industrial process control systems is that they utilize dedicated, point-to-point wired communication links to measurement sensors and control actuators using local area networks. While this paradigm to process control has been successful, chemical plant operation could substantially benefit from an efficient integration of the existing, point-to-point control networks (wired connections from each actuator or sensor to the control system using dedicated local area networks) with additional networked (wired or wireless) actuator or sensor devices that have become cheap and easy-to-install. Over the last decade, a series of papers and reports including significant industrial input has advocated this next step in the evolution of industrial process systems (e.g., [160], [31], [117], [24], [163], [99]). Today, such an augmentation in sensor information, actuation capability and network-based availability of wired and wireless data is well underway in the process industries and clearly has the potential to dramatically improve the ability of the single-process and plant-wide model-based control systems to optimize process and plant performance. Network-based communication allows for easy modification of the control strategy by rerouting signals, having redundant systems that can be

Panagiotis D. Christofides is with the Department of Chemical and Biomolecular Engineering and the Department of Electrical Engineering, University of California, Los Angeles, CA 90095-1592, USA. Riccardo Scattolini is with the Dipartimento di Elettronica e Informazione Politecnico di Milano, 20133 Milano, Italy. David Muñoz de la Peña is with the Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla, Sevilla 41092, Spain. Jinfeng Liu is with the Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095-1592, USA. Emails: `pdc@seas.ucla.edu`, `riccardo.scattolini@elet.polimi.it`, `davidmps@cartuja.us.es` and `jinfeng@ucla.edu`. P. D. Christofides is the corresponding author.

activated automatically when component failure occurs, and in general, it allows having a high-level supervisory control over the entire plant. However, augmenting existing control networks with real-time wired or wireless sensor and actuator networks challenges many of the assumptions made in the development of traditional process control methods dealing with dynamical systems linked through ideal channels with flawless, continuous communication. On one hand, the use of networked sensors may introduce asynchronous measurements or time-delays in the control loop due to the potentially heterogeneous nature of the additional measurements. On the other hand, the substantial increase of the number of decision variables, state variables and measurements, may increase significantly the computational time needed for the solution of the centralized control problem and may impede the ability of centralized control systems (particularly when nonlinear constrained optimization-based control systems like MPC are used), to carry out real-time calculations within the limits set by process dynamics and operating conditions. Furthermore, this increased dimension and complexity of the centralized control problem may cause organizational and maintenance problems as well as reduced fault-tolerance of the centralized control systems to actuator and sensor faults.

These considerations motivate the development of distributed control systems that utilize an array of controllers that carry out their calculations in separate processors yet they communicate to efficiently cooperate in achieving the closed-loop plant objectives. MPC is a natural control framework to deal with the design of coordinated, distributed control systems because of its ability to handle input and state constraints and predict the evolution of a system with time while accounting for the effect of asynchronous and delayed sampling, as well as because it can account for the actions of other actuators in computing the control action of a given set of control actuators in real-time [13]. In this paper, we provide a tutorial review of recent results in the design of distributed model predictive control systems. Our goal is to not only review the results in this area but also to provide enough algorithmic details so that the distinctions between different approaches can become quite clear and newcomers in this field can find this paper to be a useful resource. In this sense, our hope is that this paper would complement a series of recent review papers in this rapidly-evolving area [15], [129], [134].

II. Preliminaries

A. Notation

The operator $|\cdot|$ is used to denote the Euclidean norm of a vector, while we use $\|\cdot\|_Q^2$ to denote the square of a weighted Euclidean norm, i.e., $\|x\|_Q^2 = x^T Q x$ for all $x \in R^n$. A continuous function $\alpha : [0, a) \to [0, \infty)$ is said to belong to class $\mathcal{K}$ if it is strictly increasing and satisfies $\alpha(0) = 0$. A function $\beta(r, s)$ is said to be a class $\mathcal{KL}$ function if, for each fixed s, $\beta(r, s)$ belongs to class $\mathcal{K}$ functions with respect to r and, for each fixed r, $\beta(r, s)$ is decreasing with respect to s and $\beta(r, s) \to 0$ as $s \to 0$. The symbol Ω_r is used to denote the set $\Omega_r := \{x \in R^n : V(x) \le r\}$ where V is a scalar positive definite, continuous differentiable function and $V(0) = 0$, and the operator '/' denotes set subtraction, that is, $A/B := \{x \in R^n : x \in A, x \notin B\}$. The symbol $diag(v)$ denotes a square diagonal matrix whose diagonal elements are the elements of the vector v. The symbol $\oplus$ denotes the Minkowski sum. The notation t_0 indicates the initial time instant. The set $\{t_{k\ge 0}\}$ denotes a sequence of synchronous time instants such that $t_k = t_0 + k\Delta$ and $t_{k+i} = t_k + i\Delta$ where Δ is a fixed time interval and i is an integer. Similarly, the set $\{t_{a\ge 0}\}$ denotes a sequence of asynchronous time instants such that the interval between two consecutive time instants is not fixed.

B. Mathematical Models for MPC

Throughout this manuscript, we will use different types of mathematical models, both linear and nonlinear dynamic models, to present the various distributed MPC schemes. Specifically, we first consider a class of nonlinear systems composed of m interconnected subsystems where each of the subsystems can be described by the following state-space model:

$$\dot{x}_i(t) = f_i(x) + g_{si}(x)u_i(t) + k_i(x)w_i(t) \qquad (1)$$

where $x_i(t) \in R^{n_{x_i}}$, $u_i(t) \in R^{n_{u_i}}$ and $w_i(t) \in R^{n_{w_i}}$ denote the vectors of state variables, inputs and disturbances associated with subsystem i with $i = 1, \ldots, m$, respectively. The disturbance $w = [w_1^T \ \cdots \ w_i^T \cdots w_m^T]^T$ is assumed to be bounded, that is, $w(t) \in W$ with $W := \{w \in R^{n_w} : |w| \le \theta, \theta > 0\}$. The variable $x \in R^{n_x}$ denotes the state of the entire nonlinear system which is composed of the states of the m subsystems, that is $x = [x_1^T \cdots x_i^T \cdots x_m^T]^T \in R^{n_x}$. The dynamics of x can be described as follows:

$$\dot{x}(t) = f(x) + \sum_{i=1}^{m} g_i(x)u_i(t) + k(x)w(t) \qquad (2)$$

where $f = [f_1^T \cdots f_i^T \cdots f_m^T]^T$, $g_i = [\mathbf{0}^T \cdots g_{si}^T \cdots \mathbf{0}^T]^T$ with $\mathbf{0}$ being the zero matrix of appropriate dimensions, k is a matrix composed of k_i ($i = 1, \ldots, m$) and zeros whose explicit expression is omitted for brevity. The m sets of inputs are restricted to be in m nonempty convex sets $U_i \subseteq R^{m_{u_i}}$, $i = 1, \ldots, m$, which are defined as $U_i := \{u_i \in R^{n_{u_i}} : |u_i| \le u_i^{\max}\}$ where $u_i^{\max}$, $i = 1, \ldots, m$, are the magnitudes of the input constraints in an element-wise manner. We assume that f, g_i, $i = 1, \ldots, m$, and k are locally Lipschitz vector functions and that the origin is an equilibrium point of the unforced nominal system (i.e., system of Eq. 2 with $u_i(t) = 0$, $i = 1, \ldots, m$, $w(t) = 0$ for all t) which implies that $f(0) = 0$.

Finally, in addition to MPC formulations based on continuous-time nonlinear systems, many MPC algorithms have been developed for systems described by a discrete-time linear model, possibly obtained from the linearization and discretization of a nonlinear continuous-time model of the form of Eq. 1.

Specifically, the linear discrete-time counterpart of the system of Eq. 1 is:

$$x_i(k+1) = A_{ii}x_i(k) + \sum_{i \neq j} A_{ij}x_j(k) + B_i u_i(k) + w_i(k) \quad (3)$$

where k is the discrete time index and the state and control variables are restricted to be in convex, non-empty sets including the origin, i.e. $x_i \in X_i$, $u_i \in U_i$. It is also assumed that $w_i \in W_i$, where W_i is a compact set containing the origin, with $W_i = \{0\}$ in the nominal case. Subsystem j is said to be a *neighbor* of subsystem i if $A_{ij} \neq 0$.

Then the linear discrete-time counterpart of the system of Eq. 2, consisting of m-subsystems of the type of Eq.3, is:

$$x(k+1) = Ax(k) + Bu(k) + w(k) \quad (4)$$

where $x \in X = \prod_i X_i$, $u \in U = \prod_i U_i$ and $w \in W = \prod_i W_i$ are the state, input and disturbance vectors, respectively.

The systems of Eq. 1 and 3 assume that the m susbsytems are coupled through the states but not through the inputs. Another class of linear systems which has been studied in the literature in the context of DMPC are systems coupled only through the inputs, that is,

$$x_i(k+1) = A_i x_i(k) + \sum_{l=1}^{m} B_{il} u_l(k) + w_i(k) \quad (5)$$

C. Lyapunov-based control

Lyapunov-based control plays an important role in determining stability regions for the closed-loop system in some of the DMPC architectures to be discussed below. Specifically, we assume that there exists a Lyapunov-based locally Lipschitz control law $h(x) = [h_1(x) \ \dots \ h_m(x)]^T$ with $u_i = h_i(x)$, $i = 1, \dots, m$, which renders the origin of the nominal closed-loop system (i.e., system of Eq. 2 with $u_i = h_i(x)$, $i = 1, \dots, m$, and $w = 0$) asymptotically stable while satisfying the input constraints for all the states x inside a given stability region. Using converse Lyapunov theorems [94], [78], [25], this assumption implies that there exist functions $\alpha_i(\cdot)$, $i = 1, 2, 3$ of class $\mathcal{K}$ and a continuously differentiable Lyapunov function $V(x)$ for the nominal closed-loop system that satisfy the following inequalities:

$$\begin{gathered} \alpha_1(|x|) \leq V(x) \leq \alpha_2(|x|) \\ \frac{\partial V(x)}{\partial x}\left(f(x) + \sum_{i=1}^{m} g_i(x)h_i(x)\right) \leq -\alpha_3(|x|) \\ h_i(x) \in U_i, \ i = 1, \dots, m \end{gathered} \quad (6)$$

for all $x \in O \subseteq R^n$ where O is an open neighborhood of the origin. We denote the region $\Omega_\rho \subseteq O$ as the stability region of the nominal closed-loop system under the Lyapunov-based controller $h(x)$. We note that Ω_ρ is usually a level set of the Lyapunov function $V(x)$, i.e., $\Omega_\rho := \{x \in R^n : V(x) \leq \rho\}$.

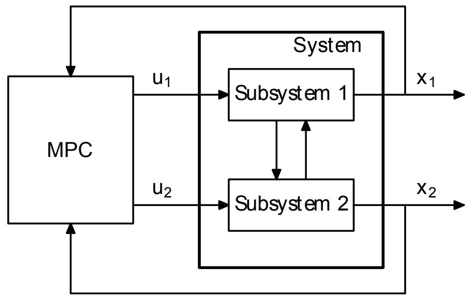

Fig. 1. Centralized MPC architecture.

III. Model Predictive Control

A. Formulation

Model predictive control (MPC) is widely adopted in industry as an effective approach to deal with large multivariable constrained control problems. The main idea of MPC is to choose control actions by repeatedly solving an online constrained optimization problem, which aims at minimizing a performance index over a finite prediction horizon based on predictions obtained by a system model [13], [85], [128]. In general, an MPC design is composed of three components:

1) A model of the system. This model is used to predict the future evolution of the system in open-loop and the efficiency of the calculated control actions of an MPC depends highly on the accuracy of the model.
2) A performance index over a finite horizon. This index will be minimized subject to constraints imposed by the system model, restrictions on control inputs and system state and other considerations at each sampling time to obtain a trajectory of future control inputs.
3) A receding horizon scheme. This scheme introduces feedback into the control law to compensate for disturbances and modeling errors.

Typically, MPC is studied from a centralized control point of view in which all the manipulated inputs of a control system are optimized with respect to an objective function in a single optimization problem. Figure 1 is a schematic of a centralized MPC architecture for a system comprised of two coupled subsystems. Consider the control of the system of Eq. 2 and assume that the state measurements of the system of Eq. 2 are available at synchronous sampling time instants $\{t_{k \geq 0}\}$, a standard MPC is formulated as follows [52]:

$$\min_{u_1, \dots, u_m \in S(\Delta)} J(t_k) \quad (7a)$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = f(\tilde{x}) + \sum_{i=1}^{m} g_i(\tilde{x}) u_i(t) \quad (7b)$$

$$u_i(t) \in U_i, \ i = 1, \dots, m \quad (7c)$$

$$\tilde{x}(t_k) = x(t_k) \quad (7d)$$

with

$$J(t_k) = \sum_{i=1}^{m} \int_{t_k}^{t_{k+N}} \left[\|\tilde{x}_i(\tau)\|^2_{Q_{ci}} + \|u_i(\tau)\|^2_{R_{ci}} \right] d\tau$$

where $S(\Delta)$ is the family of piece-wise constant functions with sampling period Δ, N is the prediction horizon, Q_{ci} and R_{ci} are strictly positive definite symmetric weighting matrices, and $\tilde{x}_i$, $i = 1, \ldots, m$, are the predicted trajectories of the nominal subsystem i with initial state $x_i(t_k)$, $i = 1, \ldots, m$, at time t_k. The objective of the MPC of Eq.7 is to achieve stabilization of the nominal system of Eq.7 at the origin, i.e., $(x, u) = (0, 0)$.

The optimal solution to the MPC optimization problem defined by Eq. 7 is denoted as $u_i^*(t|t_k)$, $i = 1, \ldots, m$, and is defined for $t \in [t_k, t_{k+N})$. The first step values of $u_i^*(t|t_k)$, $i = 1, \ldots, m$, are applied to the closed-loop system for $t \in [t_k, t_{k+1})$. At the next sampling time t_{k+1}, when new measurements of the system states $x_i(t_{k+1})$, $i = 1, \ldots, m$, are available, the control evaluation and implementation procedure is repeated. The manipulated inputs of the system of Eq. 2 under the control of the MPC of Eq. 7 are defined as follows:

$$u_i(t) = u_i^*(t|t_k),\ \forall t \in [t_k, t_{k+1}),\ i = 1, \ldots, m \tag{8}$$

which is the standard receding horizon scheme.

In the MPC formulation of Eq. 7, the constraint of Eq. 7a defines a performance index or cost index that should be minimized. In addition to penalties on the state and control actions, the index may also include penalties on other considerations; for example, the rate of change of the inputs. The constraint of Eq. 7b is the nominal model, that is, the uncertainties are supposed to be zero in the model of Eq. 2 which is used in the MPC to predict the future evolution of the process. The constraint of Eq. 7c takes into account the constraints on the control inputs, and the constraint of Eq. 7d provides the initial state for the MPC which is a measurement of the actual system state. Note that in the above MPC formulation, state constraints are not considered but can be readily taken into account.

B. Stability

It is well known that the MPC of Eq. 7 is not necessarily stabilizing. To achieve closed-loop stability, different approaches have been proposed in the literature. One class of approaches is to use infinite prediction horizons or well-designed terminal penalty terms; please see [12], [96] for surveys of these approaches. Another class of approaches is to impose stability constraints in the MPC optimization problem (e.g.,[18], [96]). There are also efforts focusing on getting explicit stabilizing MPC laws using offline computations [86]. However, the implicit nature of MPC control law makes it very difficult to explicitly characterize, a priori, the admissible initial conditions starting from where the MPC is guaranteed to be feasible and stabilizing. In practice, the initial conditions are usually chosen in an *ad hoc* fashion and tested through extensive closed-loop simulations. To address this issue, Lyapunov-based MPC (LMPC) designs have been proposed in [102], [103] which allow for an explicit characterization of the stability region and guarantee controller feasibility and closed-loop stability. Below we review various methods for ensuring closed-loop stability under MPC that are utilized in the DMPC results to be discussed in the following sections.

We start with stabilizing MPC formulations for linear discrete-time systems based on terminal weight and terminal constraints. Specifically, a standard centralized MPC is formulated as follows:

$$\min_{u(k),\ldots,u(k+N-1)} J(k) \tag{9}$$

subject to Eq. (4) with $w = 0$ and, for $j = 0, \ldots, N-1$,

$$u(k+j) \in U \,, j \geq 0 \tag{10}$$

$$x(k+j) \in X \,, j > 0 \tag{11}$$

$$x(k+N) \in X_f \tag{12}$$

with

$$J(k) = \sum_{j=0}^{N-1} [\|x(k+j)\|_Q^2 + \|u(k+j)\|_R^2] + V_f(x(k+N)) \tag{13}$$

The optimal solution is denoted $u^*(k), \ldots, u^*(k+N-1)$. At each sampling time, the corresponding first step values $u_i^*(k)$ are applied following a receding horizon approach.

The terminal set $X_f \subseteq X$ and the terminal cost V_f are used to guarantee stability properties, and can be selected according to the following simple procedure. First, assume that a linear stabilizing control law

$$u(k) = Kx(k) \tag{14}$$

is known in the unconstrained case, i.e. $A+BK$ is stable; a wise choice is to compute the gain K as the solution of an infinite-horizon linear quadratic (LQ) control problem with the same weights Q and R used in Eq. 13. Then, letting P be the solution of the Lyapunov equation

$$(A+BK)'P(A+BK) - P = -(Q + K'RK) \tag{15}$$

it is possible to set $V_f = x'Px$ and $X_f = \{x|x'Px \leq c\}$, where c is a small positive value chosen so that $u = Kx \in U$ for any $x \in X_f$. These choices implicitly guarantee a decreasing property of the optimal cost function (similar to the one explicitly expressed by the constraint of Eq. 16e below in the context of Lyapunov-based MPC), so that the origin of the state space is an asymptotically stable equilibrium with a region of attraction given by the set of the states for which a feasible solution of the optimization problem exists, see, for example, [96]. Many other choices of the design parameters guaranteeing stability properties for linear and nonlinear systems have been proposed see, for example, [97], [33], [89], [90], [50], [109], [56], [53].

In addition to stabilizing MPC formulations based on terminal weight and terminal constraints, we also review a formulation using Lyapunov function-based stability constraints since it is utilized in some of the DMPC schemes to be presented below. Specifically, we review the LMPC design proposed in [102], [103] which allows for an explicit characterization of the stability region and guarantees controller feasibility and closed-loop stability. For the predictive control of the system of Eq. 2, the LMPC is designed based on an existing explicit control law $h(x)$ which is able to stabilize the closed-loop system and satisfies the conditions of Eq. 6. The formulation of the LMPC is as follows:

$$\min_{u_1,\ldots,u_m \in S(\Delta)} J(t_k) \tag{16a}$$

$$\text{s.t.}\quad \dot{\tilde{x}}(t) = f(\tilde{x}) + \sum_{i=1}^{m} g_i(\tilde{x})u_i(t) \tag{16b}$$

$$u(t) \in U \tag{16c}$$

$$\tilde{x}(t_k) = x(t_k) \tag{16d}$$

$$\frac{\partial V(x(t_k))}{\partial x} g_i(x(t_k))u_i(t_k) \leq \frac{\partial V(x(t_k))}{\partial x} g_i(x(t_k))h_i(x(t_k)) \tag{16e}$$

where $V(x)$ is a Lyapunov function associated with the nonlinear control law $h(x)$. The optimal solution to this LMPC optimization problem is denoted as $u_i^{l,*}(t|t_k)$ which is defined for $t \in [t_k, t_{k+N})$. The manipulated input of the system of Eq. 2 under the control of the LMPC of Eq. 16 is defined as follows:

$$u_i(t) = u_i^{l,*}(t|t_k), \quad \forall t \in [t_k, t_{k+1}) \tag{17}$$

which implies that this LMPC also adopts a standard receding horizon strategy.

In the LMPC defined by Eq. 16, the constraint of Eq. 16e guarantees that the value of the time derivative of the Lyapunov function, $V(x)$, at time t_k is smaller than or equal to the value obtained if the nonlinear control law $u = h(x)$ is implemented in the closed-loop system in a sample-and-hold fashion. This is a constraint that allows one to prove (when state measurements are available every synchronous sampling time) that the LMPC inherits the stability and robustness properties of the nonlinear control law $h(x)$ when it is applied in a sample-and-hold fashion. Specifically, one of the main properties of the LMPC of Eq. 16 is that it possesses the same stability region Ω_ρ as the nonlinear control law $h(x)$, which implies that the origin of the closed-loop system is guaranteed to be stable and the LMPC is guaranteed to be feasible for any initial state inside Ω_ρ when the sampling time Δ and the disturbance upper bound θ are sufficiently small. The stability property of the LMPC is inherited from the nonlinear control law $h(x)$ when it is applied in a sample-and-hold fashion; please see [28], [110] for results on sampled-data systems. The feasibility property of the LMPC is also guaranteed by the nonlinear control law $h(x)$ since $u = h(x)$ is a feasible solution to the optimization problem of Eq. 16 (see also [102], [103], [92] for detailed results on this issue). The main advantage of the LMPC approach with respect to the nonlinear control law $h(x)$ is that optimality considerations can be taken explicitly into account (as well as constraints on the inputs and the states [103]) in the computation of the control actions within an online optimization framework while improving the closed-loop performance of the system. We finally note that since the closed-loop stability and feasibility of the LMPC of Eq. 16 are guaranteed by the nonlinear control law $h(x)$, it is unnecessary to use a terminal penalty term in the cost index and the length of the horizon N does not affect the stability of the closed-loop system but it affects the closed-loop performance.

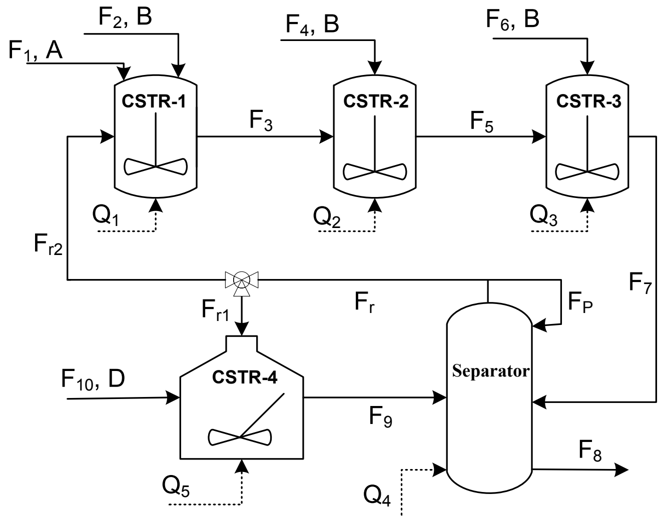

Fig. 2. Alkylation of benzene with ethylene process.

C. Alkylation of benzene with ethylene process example

We now introduce a chemical process network example to discuss the selection of the control configurations in the context of the various MPC formulations. The process considered is the alkylation of benzene with ethylene and consists of four continuously stirred tank reactors (CSTRs) and a flash tank separator, as shown in Fig. 2. The CSTR-1, CSTR-2 and CSTR-3 are in series and involve the alkylation of benzene with ethylene. Pure benzene is fed through stream F_1 and pure ethylene is fed through streams F_2, F_4 and F_6. Two catalytic reactions take place in CSTR-1, CSTR-2 and CSTR-3. Benzene (A) reacts with ethylene (B) and produces the required product ethylbenzene (C) (reaction 1); ethylbenzene can further react with ethylene to form 1,3-diethylbenzene (D) (reaction 2) which is the byproduct. The effluent of CSTR-3, including the products and leftover reactants, is fed to a flash tank separator, in which most of benzene is separated overhead by vaporization and condensation techniques and recycled back to the plant, and the bottom product stream is removed. A portion of the recycle stream F_{r2} is fed back to CSTR-1 and another portion of the recycle stream F_{r1} is fed to CSTR-4 together with an additional feed stream F_{10} which contains 1,3-diethylbenzene from another distillation process that we do not explicitly consider in this example. In CSTR-4, reaction 2 and a catalyzed transalkylation reaction in which 1,3-diethylbenzene reacts with benzene to produce ethylbenzene (reaction 3) take place. All chemicals left from CSTR-4 eventually pass into the separator. All the materials in the reactions are in liquid phase due to high pressure.

The control objective is to stabilize the process at a desired operating steady-state and achieve an optimal level of closed-loop performance. To accomplish the control objective, we

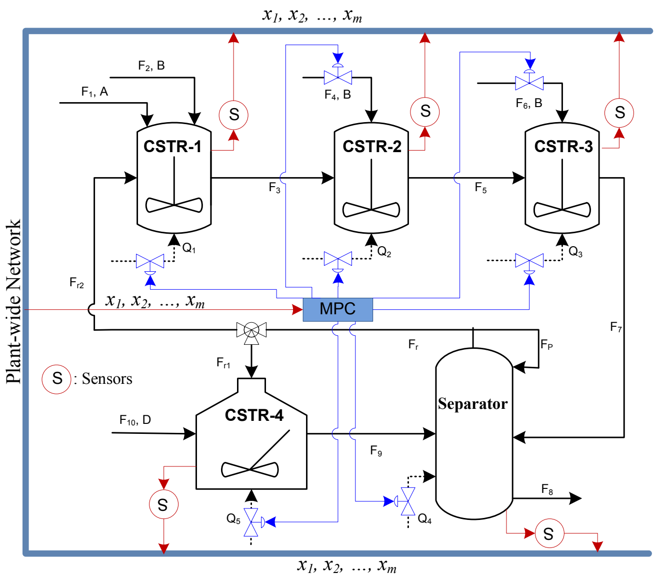

Fig. 3. Centralized MPC configuration for the alkylation of benzene process.

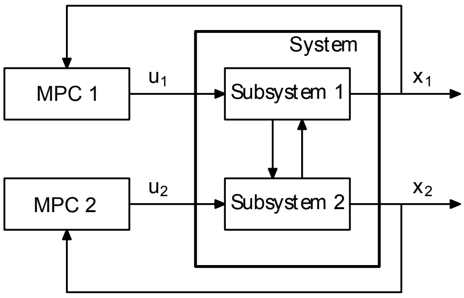

Fig. 4. Decentralized MPC architecture.

may manipulate the five heat inputs/removals, Q_1, Q_2, Q_3, Q_4, Q_5, as well as the two ethylene input flow rates, F_4 and F_6. For a centralized MPC architecture, all the inputs will be optimized in one optimization problem as shown in Fig. 3.

IV. Decentralized Model Predictive Control

While there are some important reviews on decentralized control (e.g., [132], [139], [9], [140]), in this section we focus on results pertaining to decentralized MPC. The key feature of a decentralized control framework is that there is no communication between the different local controllers. A schematic of a decentralized MPC architecture with two subsystems is shown in Fig. 4. It is well known that strong interactions between different subsystems may prevent one from achieving stability and desired performance with decentralized control (e.g., [32], [158]). In general, in order to achieve closed-loop stability as well as performance in the development of decentralized MPC algorithms, the interconnections between different subsystems are assumed to be weak and are considered as disturbances which can be compensated through feedback so they are not involved in the controller formulation explicitly.

Consider the control of the system of Eq. 2 and assume that the state measurements of the system of Eq. 2 are available at synchronous sampling time instants $\{t_{k\geq 0}\}$, a typical decentralized MPC is formulated as follows:

$$\min_{u_i \in S(\Delta)} J_i(t_k) \tag{18a}$$

$$\text{s.t. } \dot{\tilde{x}}_i(t) = f_i(\tilde{x}_{i-}(t)) + g_{si}(\tilde{x}_{i-})u_i(t) \tag{18b}$$

$$u_i(t) \in U_i \tag{18c}$$

$$\tilde{x}_i(t_k) = x_i(t_k) \tag{18d}$$

with

$$J_i(t_k) = \int_{t_k}^{t_{k+N}} \left[\|\tilde{x}_i(\tau)\|_{Q_{ci}}^2 + \|u_i(\tau)\|_{R_{ci}}^2 \right] d\tau$$

where $x_{i-} = [0 \cdots x_i \cdots 0]^T$, J_i is the cost function used in each individual local controller based on its local subsystem states and control inputs.

In [91], a decentralized MPC algorithm for nonlinear discrete time systems subject to decaying disturbances was presented. No information is exchanged between the local controllers and the stability of the closed-loop system relies on the inclusion of a contractive constraint in the formulation of each of the decentralized MPCs. In the design of the decentralized MPC, the effects of interconnections between different subsystems are considered as perturbation terms whose magnitude depend on the norm of the system states. In [124], the stability of a decentralized MPC is analyzed from an input-to-state stability (ISS) point of view. In [2], a decentralized MPC algorithm was developed for large-scale linear processes subject to input constraints. In this work, the global model of the process is approximated by several (possibly overlapping) smaller subsystem models which are used for local predictions and the degree of decoupling among the subsystem models is a tunable parameter in the design. In [3], possible date packet dropouts in the communication between the distributed controllers were considered in the context of linear systems and their influence on the closed-loop system stability was analyzed.

To develop coordinated decentralized control systems, the dynamic interaction between different units should be considered in the design of the control systems. This problem of identifying dynamic interactions between units was studied in [55].

Within process control, another important work on the subject of decentralized control includes the development of a quasi-decentralized control framework for multi-unit plants that achieves the desired closed-loop objectives with minimal cross communication between the plant units under state feedback control [146]. In this work, the idea is to incorporate in the local control system of each unit a set of dynamic models that provide an approximation of the interactions between the different subsystems when local subsystem states are not exchanged between different subsystems and to update the state of each model using states information exchanged when communication is re-established.

In general, the overall closed-loop performance under a decentralized control system is limited because of the limitation

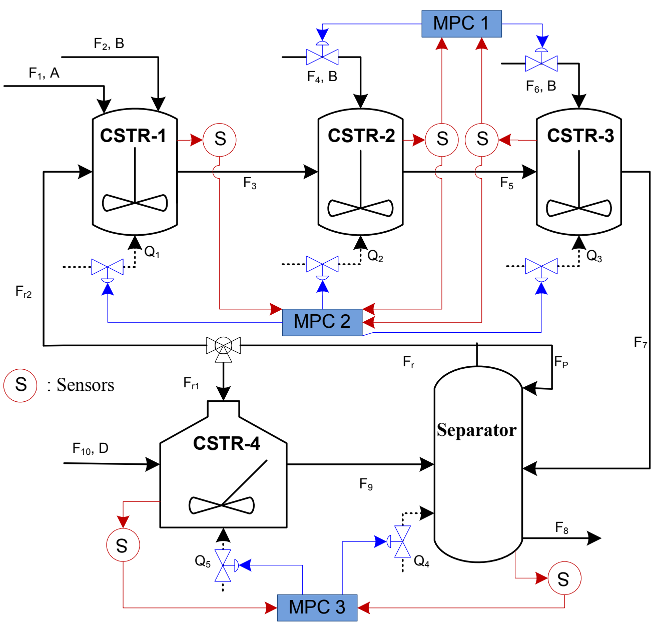

Fig. 5. Decentralized MPC configuration for the alkylation of benzene with ethylene process.

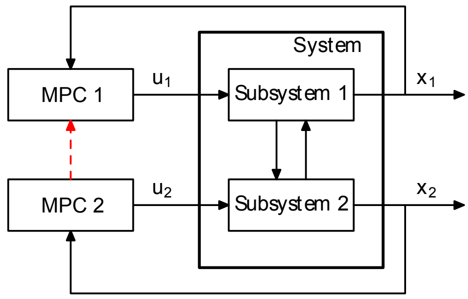

Fig. 6. Sequential DMPC architecture.

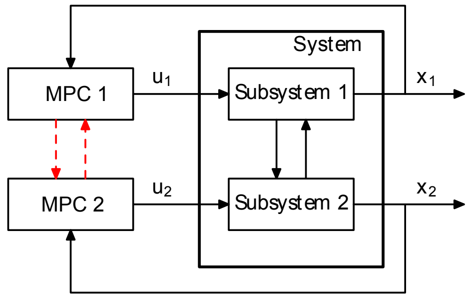

Fig. 7. Parallel DMPC architecture.

on the available information and the lack of communication between different controllers [29]. This leads us to the design of model predictive control architectures in which different MPCs coordinate their actions through communication to exchange subsystem state and control action information.

A. Alkylation of benzene with ethylene process example (Cont')

For the alkylation of benzene process, we may design three decentralized MPCs to manipulate the seven inputs as shown in Fig. 5. In this decentralized control configuration, the first controller (MPC 1) is used to compute the values of Q_1, Q_2 and Q_3, the second distributed controller (MPC 2) is used to compute the values of Q_4 and Q_5, and the third controller (MPC 3) is used to compute the values of F_4 and F_6. The three controllers make their decisions independently and do not exchange any information.

V. Distributed Model Predictive Control

To achieve better closed-loop control performance, some level of communication may be established between the different controllers, which leads to distributed model predictive control (DMPC). With respect to available results in this direction, several DMPC methods have been proposed as well as some important review articles [129], [134] have been written which primarily focus the review of the various DMPC schemes at a conceptual level. With respect to the DMPC algorithms available in the literature, a classification can be made according to the topology of the communication network, the different communication protocols used by local controllers, and the cost function considered in the local controller optimization problem [134]. In the following, we will classify the different algorithms based on the cost function used in the local controller optimization problem as used in [129]. Specifically, we will refer to the distributed algorithms in which each local controller optimizes a local cost function as *non-cooperative* DMPC algorithms, and refer to the distributed algorithms in which each local controller optimizes a global cost function as *cooperative* DMPC algorithms.

A. Non-Cooperative DMPC

In [130], a DMPC algorithm was proposed for a class of decoupled systems with coupled constraints. This class of systems captures an important class of practical problems, including, for example, maneuvering a group of vehicles from one point to another while maintaining relative formation and/or avoiding collisions. In [130], the distributed controllers are evaluated in sequence which means that controller $i + 1$ is evaluated after controller i has been evaluated or vice versa. A *sequential* DMPC architecture with two local controllers is shown in Fig. 6. An extension of this work [150] proposes the use of the robust design method described in [98] for DMPC.

In the majority of the algorithms in the category of non-cooperative DMPC, the distributed controllers are evaluated in *parallel* i.e., at the same time. The controllers may be only evaluated once (*non-iterative*) or iterate (*iterative*) to achieve a solution at a sampling time. A *parallel* DMPC architecture with two local controllers is shown in Fig. 7.

Many parallel DMPC algorithms in the literature belong to the *non-iterative* category. In [15], a DMPC algorithm was proposed for a class of discrete-time linear systems. In this work, a stability constraint is included in the problem formulation and the stability can be verified a-posteriori with an analysis of the resulting closed-loop system. In [68], DMPC for systems with dynamically decoupled subsystems (a class of systems of relevance in the context of multi-agents systems) where the cost function and constraints couple the dynamical behavior of the system. The coupling in the system is described using a graph in which each subsystem is a node. It is assumed that each subsystem can exchange information with its neighbors (a subset of other subsystems). Based on the results of [68], a DMPC framework was constructed for control and coordination of autonomous vehicle teams [69].

In [64], a DMPC scheme for linear systems coupled only through the state is considered, while [38] deals with the problem of distributed control of dynamically decoupled nonlinear systems coupled by their cost function. This method is extended to the case of dynamically coupled nonlinear systems in [36] and applied as a distributed control strategy in the context of supply chain optimization in [37]. In this implementation, the agents optimize locally their own policy, which is communicated to their neighbors. The stability is assured through a compatibility constraint: the agents commit themselves not to deviate too far in their state and input trajectories from what their neighbors believe they plan to do. In [100] another iterative implementation of a similar DMPC scheme was applied together with a distributed Kalman filter to a quadruple tank system. Finally, in [76] the Shell benchmark problem is used to test a similar algorithm. Note that all these methods lead in general to Nash equilibria as long as the cost functions of the agents are selfish.

1) A noncooperative DMPC algorithm: As an example of a noncooperative DMPC algorithm for discrete-time systems described by Eq. 3, we now synthetically describe the method recently proposed in [46] relying on the "tube-based" approach developed in [98] for the design of robust MPC. The rationale is that each subsystem i transmits to its neighbors its planned state reference trajectory $\tilde{x}_i(k+j)$, $j = 1, ..., N$, over the prediction horizon and "guarantees" that, for all $j \geq 0$, its actual trajectory lies in a neighborhod of $\tilde{x}_i$, i.e. $x_i(k+j) \in \tilde{x}_i(k+j) \oplus \mathcal{E}_i$, where $\mathcal{E}_i$ is a compact set including the origin. In this way, Eq. (3) can be written as

$$x_i(k+1) = A_{ii}x_i(k) + B_iu_i(k) + \sum_j A_{ij}\tilde{x}_j(k) + w_i(k) \tag{19}$$

where $w_i(k) = \sum_j A_{ij}(x_j(k) - \tilde{x}_j(k)) \in W_i$ is a bounded disturbance, $W_i = \bigoplus_j A_{ij}\mathcal{E}_i$ and the term $\sum_j A_{ij}\tilde{x}_j(k)$ can be interpreted as an input, known in advance over the prediction horizon. Note that in this case, we assume that the only disturbance of each model is due to the mismatch between the planned and real state trajectories.

From Eq. (19), the i-th subsystem nominal model is defined as

$$\hat{x}_i(k+1) = A_{ii}\hat{x}_i(k) + B_i\hat{u}_i(k) + \sum_j A_{ij}\tilde{x}_j(k) \tag{20}$$

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

$$u_i(k) = \hat{u}_i(k) + K_i(x_i(k) - \hat{x}_i(k)) \tag{21}$$

From Eq. (19) and Eq. (21) and letting $z_i(k) = x_i(k) - \hat{x}_i(k)$, we obtain:

$$z_i(k+1) = (A_{ii} + B_iK_i)z_i(k) + w_i(k) \tag{22}$$

where $w_i \in W_i$. Since W_i is bounded and $A_{ii} + B_iK_i$ is stable, there exists a robust positively invariant set Z_i for Eq. (22) such that, for all $z_i(k) \in Z_i$ and $w_i(k) \in W_i$, then $z_i(k+1) \in Z_i$. Given Z_i and assuming that there exist neighborhoods of the origin E_i such that $E_i \oplus Z_i \subseteq \mathcal{E}_i$, at any time instant k, the i-th subsystem computes the value of $\hat{u}_i(k)$ in Eq. 21 as the solution of

$$\min_{\hat{x}_i(k), \hat{u}_i(k), \ldots, \hat{u}_i(k+N-1)} J_i(k) \tag{23}$$

subject to Eq. (20) and, for $j = 0, \ldots, N-1$,

$$x_i(k) - \hat{x}_i(k) \in Z_i \tag{24}$$

$$\hat{x}_i(k+j) - \tilde{x}_i(k+j) \in E_i \tag{25}$$

$$\hat{x}_i(k+j) \in \hat{X}_i \subseteq X_i \ominus Z_i \tag{26}$$

$$\hat{u}_i(k+j) \in \hat{U}_i \subseteq U_i \ominus K_iZ_i \tag{27}$$

$$\hat{x}_i(k+N) \in \hat{X}_{fi} \tag{28}$$

In this problem,

$$J_i(k) = \sum_{j=0}^{N-1} [\|x_i(k+j)\|^2_{Q_i} + \|u_i(k+j)\|^2_{R_i}] + \|x(k+N)\|^2_{P_i} \tag{29}$$

and the restricted constraints given by Eqs. 24-27 are used to ensure that the difference between x_i and $\tilde{x}_i$ is effectively limited, as initially stated, while a proper choice of the weights Q_i, R_i, P_i and of the terminal set X_{fi} guarantee the stabilizing properties of the method, please see [46] for details. Finally, with the optimal solution at time k, it is also possible to compute the predicted value $\hat{x}_i(k+N)$, which is used to incrementally define the reference trajectory of the state to be used at the next time instant $k+1$, i.e. $\tilde{x}_i(k+N) = \hat{x}_i(k+N)$.

B. Cooperative DMPC

The key feature of cooperative DMPC is that in each of the local controllers, the same global cost function is optimized. In recent years, many efforts have been made to develop cooperative DMPC for linear and nonlinear systems.

The idea of cooperative DMPC was first introduced in [154] and later developed in [129]. In the latter work, a set of linear systems coupled through the inputs of the type presented in Eq. 5 were considered.

In cooperative DMPC each controller takes into account the effects of its inputs on the entire plant through the use of a centralized cost function. At each iteration, each controller optimizes its own set of inputs assuming that the rest of the inputs of its neighbors are fixed to the last agreed value. Subsequently, the controllers share the resulting optimal trajectories and a final optimal trajectory is computed at each sampling time as a weighted sum of the most recent optimal trajectories with the optimal trajectories computed at the last sampling time.

The cooperative DMPCs use the following implementation strategy:

1. At k, all the controllers receive the full state measurement $x(k)$ from the sensors.
2. At iteration c ($c \geq 1$):
   2.1. Each controller evaluates its own future input trajectory based on $x(k)$ and the latest received input trajectories of all the other controllers (when $c = 1$, initial input guesses obtained from the shifted latest optimal input trajectories are used).
   2.2. The controllers exchange their future input trajectories. Based on all the input trajectories, each controller calculates the current decided set of inputs trajectories u^c.
3. If a termination condition is satisfied, each controller sends its entire future input trajectory to its actuators; if the termination condition is not satisfied, go to Step 2 ($c \leftarrow c + 1$).
4. When a new measurement is received, go to Step 1 ($k \leftarrow k + 1$).

At each iteration, each controller solves the following optimization problem:

$$\min_{u_i(k),\ldots,u_i(k+N-1)} J(k) \tag{30}$$

subject to Eq. (4) with $w = 0$ and, for $j = 0, \ldots, N-1$,

$$u_i(k+j) \in U_i \,, j \geq 0 \tag{31}$$

$$u_l(k+j) = u_l(k+j)^{c-1} \,, \forall l \neq i \tag{32}$$

$$x(k+j) \in X \,, j > 0 \tag{33}$$

$$x(k+N) \in X_f \tag{34}$$

with

$$J(k) = \sum_i J_i(k) \tag{35}$$

and

$$J_i(k) = \sum_{j=0}^{N-1} [\|x_i(k+j)\|^2_{Q_i} + \|u_i(k+j)\|^2_{R_i}] + \|x(k+N)\|^2_{P_i} \tag{36}$$

Note that each controller must have knowledge of the full system dynamics and of the overall objective function.

After the controllers share the optimal solutions $u_i(k+j)^*$, the optimal trajectory at iteration c, $u_i(k+j)^c$, is obtained from a convex combination between the last optimal solution and the current optimal solution of the MPC problem of each controller, that is,

$$u_i(k+j)^c = \alpha_i u_i(k+j)^{c-1} + (1 - \alpha_i) u_i(k+j)^*$$

where α_i are the weighting factors for each agent. This distributed optimization is of the Gauss-Jacobi type.

In [154], [143], an iterative cooperative DMPC algorithm was designed for linear systems. It was proven that through multiple communications between distributed controllers and using system-wide control objective functions, stability of the closed-loop system can be guaranteed for linear systems, and the closed-loop performance converges to the corresponding centralized control system as the iteration number increases. A design method to choose the stability constraints and the cost function is given that guarantees feasibility (given an initial feasible guess), convergence and optimality (if the constraints of the inputs are not coupled) of the resulting distributed optimization algorithm. In addition, the stability properties of the resulting closed-loop system, output feedback implementations and coupled constraints are also studied.

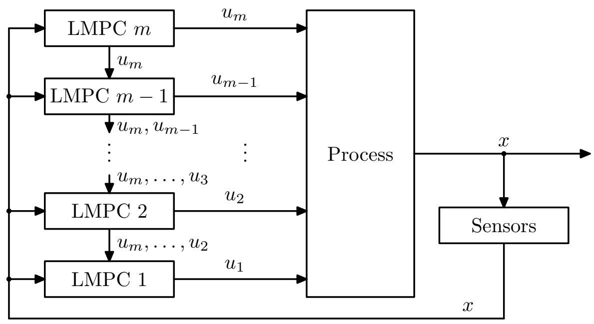

Fig. 8. Sequential DMPC architecture using LMPC [80].

The properties of cooperative DMPC are strongly based on convexity. In [144], the results were extended to include nonlinear systems and the resulting non-convex optimization problems without guaranteed convergence of the closed-loop performance to the corresponding centralized control system. Two cooperative and iterative DMPC algorithms for cascade processes have been described in [162], where the performance index minimized by each agent includes the cost functions of its neighborhoods, communication delays are considered and stability is proven in the unconstrained case. In addition to these results, recent efforts [82], [80] have focused on the development of Lyapunov-based sequential and iterative, cooperative DMPC algorithms for nonlinear systems with well-characterized regions of closed-loop stability. Below we discuss these DMPC algorithms.

1) Sequential DMPC: In [82], [80], a sequential DMPC architecture shown in Figure 8 for fully coupled nonlinear systems was developed based on the assumption that the full system state feedback is available to all the distributed controllers at each sampling time. In the proposed sequential DMPC, for each set of the control inputs u_i, a Lyapunov-based MPC (LMPC), denoted LMPC i, is designed. The distributed LMPCs use the following implementation strategy:

1) At t_k, all the LMPCs receive the state measurement $x(t_k)$ from the sensors.
2) For $j = m$ to 1
   2.1. LMPC j receives the entire future input trajectories of u_i, $i = m, \ldots, j+1$, from LMPC $j+1$ and evaluates the future input trajectory of u_j based on $x(t_k)$ and the received future input

trajectories.

2.2. LMPC j sends the first step input value of u_j to its actuators and the entire future input trajectories of u_i, $i = m, \dots, j$, to LMPC $j-1$.

3) When a new measurement is received ($k \leftarrow k+1$), go to Step 1.

In this architecture, each LMPC only sends its future input trajectory and the future input trajectories it received to the next LMPC (i.e., LMPC j sends input trajectories to LMPC $j-1$). This implies that LMPC j, $j = m, \dots, 2$, does not have any information about the values that u_i, $i = j-1, \dots, 1$ will take when the optimization problems of the LMPCs are designed. In order to make a decision, LMPC j, $j = m, \dots, 2$ must assume trajectories for u_i, $i = j-1, \dots, 1$, along the prediction horizon. To this end, an explicit nonlinear control law $h(x)$ which can stabilize the closed-loop system asymptotically is used. In order to inherit the stability properties of the controller $h(x)$, a Lyapunov function based constraint is incorporated in each LMPC to guarantee a given minimum contribution to the decrease rate of the Lyapunov function $V(x)$. Specifically, the design of LMPC j, $j = 1, \dots, m$, is based on the following optimization problem:

$$\min_{u_j \in S(\Delta)} J(t_k) \tag{37a}$$

$$\text{s.t.} \dot{\tilde{x}}(t) = f(\tilde{x}(t)) + \sum_{i=1}^{m} g_i(\tilde{x}(t)) u_i \tag{37b}$$

$$u_i(t) = h_i(\tilde{x}(t_{k+l})), i = 1, \dots, j-1,$$
$$\forall t \in [t_{k+l}, t_{k+l+1}), l = 0, ..., N-1 \tag{37c}$$

$$u_i(t) = u^*_{s,i}(t|t_k), i = j+1, \dots, m \tag{37d}$$

$$u_j(t) \in U_j \tag{37e}$$

$$\tilde{x}(t_k) = x(t_k) \tag{37f}$$

$$\frac{\partial V(x(t_k))}{\partial x} g_j(x(t_k)) u_j(t_k) \leq \frac{\partial V(x(t_k))}{\partial x} g_j(x(t_k)) h_j(x(t_k)). \tag{37g}$$

In the optimization problem of Eq. 37, $u^*_{s,i}(t|t_k)$ denotes the optimal future input trajectory of u_i obtained by LMPC i evaluated before LMPC j. The constraint of Eq. 37c defines the value of the inputs evaluated after u_j (i.e., u_i with $i = 1, \dots, j-1$); the constraint of Eq. 37d defines the value of the inputs evaluated before u_j (i.e., u_i with $i = j+1, \dots, m$); the constraint of Eq. 37g guarantees that the contribution of input u_j to the decrease rate of the time derivative of the Lyapunov function $V(x)$ at the initial evaluation time (i.e., at t_k), if $u_j = u^*_{s,j}(t_k|t_k)$ is applied, is bigger than or equal to the value obtained when $u_j = h_j(x(t_k))$ is applied. This constraint allows proving the closed-loop stability properties of this DMPC [82], [80].

2) Iterative DMPC: In [80], a Lyapunov-based iterative DMPC algorithm shown in Figure 9 was proposed for coupled nonlinear systems. The implementation strategy of this iterative DMPC is as follows:

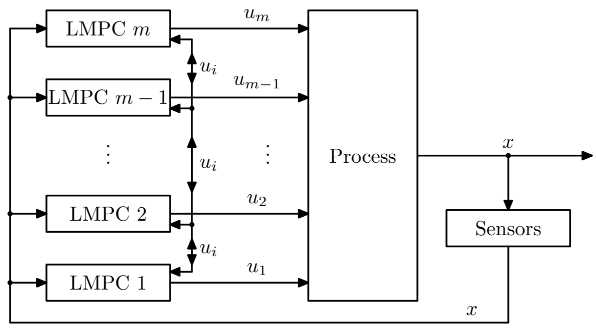

Fig. 9. Iterative DMPC architecture using LMPC [80].

1. At t_k, all the LMPCs receive the state measurement $x(t_k)$ from the sensors and then evaluate their future input trajectories in an iterative fashion with initial input guesses generated by $h(\cdot)$.
2. At iteration c ($c \geq 1$):
   2.1. Each LMPC evaluates its own future input trajectory based on $x(t_k)$ and the latest received input trajectories of all the other LMPCs (when $c = 1$, initial input guesses generated by $h(\cdot)$ are used).
   2.2. The controllers exchange their future input trajectories. Based on all the input trajectories, each controller calculates and stores the value of the cost function.
3. If a termination condition is satisfied, each controller sends its entire future input trajectory corresponding to the smallest value of the cost function to its actuators; if the termination condition is not satisfied, go to Step 2 ($c \leftarrow c+1$).
4. When a new measurement is received, go to Step 1 ($k \leftarrow k+1$).

Note that at the initial iteration, all the LMPCs use $h(x)$ to estimate the input trajectories of all the other controllers. Note also that the number of iterations c can be variable and it does not affect the closed-loop stability of the DMPC architecture presented in this subsection. For the iterations in this DMPC architecture, there are different choices of the termination condition. For example, the number of iterations c may be restricted to be smaller than a maximum iteration number $c_{\max}$ (i.e., $c \leq c_{\max}$) and/or the iterations may be terminated when the difference of the performance or the solution between two consecutive iterations is smaller than a threshold value and/or the iterations maybe terminated when a maximum computational time is reached. In order to proceed, we define $\hat{x}(t|t_k)$ for $t \in [t_k, t_{k+N})$ as the nominal sampled trajectory of the system of Eq. 2 associated with the feedback control law $h(x)$ and sampling time Δ starting from $x(t_k)$. This nominal sampled trajectory is obtained by integrating recursively the following differential equation:

$$\dot{\hat{x}}(t|t_k) = f(\hat{x}(t|t_k)) + \sum_{i=1}^{m} g_i(\hat{x}(t|t_k)) h_i(\hat{x}(t_{k+l}|t_k)),$$
$$\forall \tau \in [t_{k+l}, t_{k+l+1}), l = 0, \dots, N-1. \tag{38}$$

Based on $\hat{x}(t|t_k)$, we can define the following variable:

$$\begin{aligned} u_{n,j}(t|t_k) &= h_j(\hat{x}(t_{k+l}|t_k)), j = 1, \ldots, m, \\ & \forall \tau \in [t_{k+l}, t_{k+l+1}), l = 0, \ldots, N-1 \end{aligned} \quad (39)$$

which will be used as the initial guess of the trajectory of u_j.

The design of the LMPC j, $j = 1, \ldots, m$, at iteration c is based on the following optimization problem:

$$\min_{u_j \in S(\Delta)} J(t_k) \quad (40a)$$

$$\text{s.t.} \dot{\tilde{x}}(t) = f(\tilde{x}(t)) + \sum_{i=1}^{m} g_i(\tilde{x}(t))u_i \quad (40b)$$

$$u_i(t) = u_{p,i}^{*,c-1}(t|t_k), \forall i \neq j \quad (40c)$$

$$u_j(t) \in U_j \quad (40d)$$

$$\tilde{x}(t_k) = x(t_k) \quad (40e)$$

$$\begin{aligned} &\frac{\partial V(x(t_k))}{\partial x} g_j(x(t_k))u_j(t_k) \\ &\leq \frac{\partial V(x(t_k))}{\partial x} g_j(x(t_k))h_j(x(t_k)) \end{aligned} \quad (40f)$$

where $u_{p,i}^{*,c-1}(t|t_k)$ is the optimal input trajectories at iteration $c-1$.

In general, there is no guaranteed convergence of the optimal cost or solution of an iterated DMPC to the optimal cost or solution of a centralized MPC for general nonlinear constrained systems because of the non-convexity of the MPC optimization problems. However, with the above implementation strategy of the iterative DMPC presented in this section, it is guaranteed that the optimal cost of the distributed optimization of Eq. 40 is upper bounded by the cost of the Lyapunov-based controller $h(\cdot)$ at each sampling time.

Note that in the case of linear systems, the constraint of Eq. 40f is linear with respect to u_j and it can be verified that the optimization problem of Eq. 40 is convex. The input given by LMPC j of Eq. 40 at each iteration may be defined as a convex combination of the current optimal input solution and the previous one, for example,

$$u_{p,j}^{c}(t|t_k) = \sum_{i=1}^{m, i \neq j} w_i u_{p,j}^{c-1}(t|t_k) + w_j u_{p,j}^{*,c}(t|t_k) \quad (41)$$

where $\sum_{i=1}^{m} w_i = 1$ with $0 < w_i < 1$, $u_{p,j}^{*,c}$ is the current solution given by the optimization problem of Eq. 40 and $u_{p,j}^{c-1}$ is the convex combination of the solutions obtained at iteration $c-1$. By doing this, it is possible to prove that the optimal cost of the distributed LMPC of Eq. 40 converges to the one of the corresponding centralized control system [11], [143], [26].

3) DMPC based on agent negotiation: We review next a line of work on DMPC algorithms which adopt an *iterative* approach for constrained linear systems coupled through the inputs [87], [88]. Figure 10 shows a scheme of this class of controllers. Note that there is one agent for each subsystem and that the number of controlled inputs may differ from the number of subsystems.

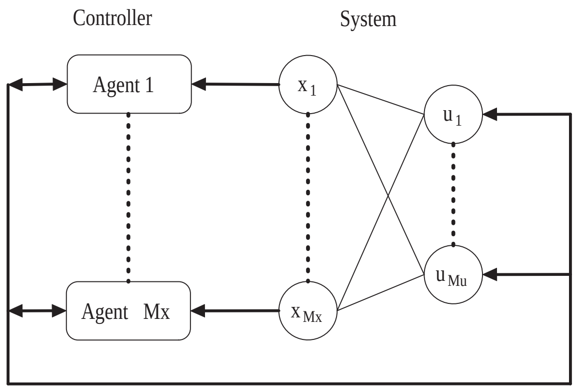

Fig. 10. DMPC based on agent negotiation.

In this class of controllers, the controllers (agents, in general) do not have any knowledge of the dynamics of any of its neighbors, but can communicate freely among them in order to reach an agreement. The proposed strategy is based on negotiation between the agents. At each sampling time, following a given protocol, agents make proposals to improve an initial feasible solution on behalf of their local cost function, state and model. These proposals are accepted if the global cost improves the cost corresponding to the current solution.

The cooperative DMPCs use the following implementation strategy:

1. At k, each one of the controllers receives its local state measurement $x_i(k)$ from its sensors and u^d is obtained shifting the decided input trajectory at time step $k-a$.
2. At iteration c ($c \geq 1$):
   - 2.1. One agent evaluates and sends a proposal to its neighbors.
   - 2.2. Each neighbor evaluates the cost increment of applying the proposal instead of the current solution u^d and sends this cost increment to the agent making the proposal.
   - 2.3. The agent making the proposal evaluates the total increment of the cost function obtained from the information received and decides the new value of u^d.
   - 2.4. The agent making the proposal communicates the decision to its neighbors.
3. If a termination condition is satisfied, each controller sends its entire future input trajectory to its actuators; if the termination condition is not satisfied, go to Step 2 ($c \leftarrow c+1$).
4. When a new measurement is received, go to Step 1 ($k \leftarrow k+1$).

Several proposals can be evaluated in parallel as long as they do not involve the same set of agents; that is, at any given time an agent can only evaluate a single proposal.

In order to generate a proposal, agent i minimizes its own local cost function J_i solving the following optimization problem:

$$\min_{u(k),\ldots,u(k+N-1)} J_i(k) \tag{42}$$

subject to Eq. (5) with $w_i = 0$ and, for $j = 0, \ldots, N-1$,

$$u_l(k+j) \in U_l\,, l \in n_{prop} \tag{43}$$
$$u_l(k+j) = u_l(k+j)^d\,, \forall l \notin n_{prop} \tag{44}$$
$$x_i(k+j) \in X_i\,, j > 0 \tag{45}$$
$$x_i(k+N) \in X_{fi} \tag{46}$$

where the $J_i(k)$ cost function depends on the predicted trajectory of x_i and the inputs which affect it. In this optimization problem, agent i optimizes over a set n_{prop} of inputs that affect its dynamics. The rest of inputs are set to the currently accepted solution $u_l(k+j)^d$.

Each agent l who is affected by the proposal of agent i evaluates the predicted cost corresponding to the proposed solution. To do so, the agent calculates the difference between the cost of the new proposal and the cost of the current accepted proposal. This information is sent to agent i, which can then evaluate the total cost of its proposal, that is, $J(k) = \sum_i J_i(k)$, to make a cooperative decision on the future inputs trajectories. If the cost improves the currently accepted solution, then $u_l(k+j)^d = u_l(k+j)^*$ for all $l \in n_{prop}$, else the proposal is discarded.

With an appropriate design of the objective functions, the terminal region constraints and assuming that an initial feasible solution is at hand, this controller can be shown to provide guaranteed stability of the resulting closed-loop system.

C. Distributed optimization

Starting from the seminal contributions reported in [101], [49], many efforts have been devoted to develop methods for the decomposition of a large optimization problem into a number of smaller and more tractable ones. Methods such as primal or dual decomposition are based on this idea; an extensive review of this kind of algorithms can be found in [11]. Dual decomposition has been used for DMPC in [125], while other augmented lagrangian formulations were proposed in [113] and applied to the control of irrigation canals in [116] and to traffic networks, see [14], [39]. In the MPC framework, algorithms based on this approach have also been described in [67], [20], [21]. A different gradient-based distributed dynamic optimization method was proposed in [136], [137] and applied to an experimental four tanks plant in [6]. The method of [136], [137] is based on the exchange of sensitivities. This information is used to modify the local cost function of each agent adding a linear term which partially allow to consider the other agents' objectives.

In order to present the basic idea underlying the application of the popular dual decomposition approach in the context of MPC, consider the set of systems of Eq. 3 in nominal conditions ($w_i = 0$) and the following (unconstrained) problem

$$\min_{u(k),\ldots,u(k+N-1)} J(k) = \sum_{i=1}^{m} J_i(k) \tag{47}$$

where

$$J_i(k) = \sum_{j=0}^{N-1} [\|x_i(k+j)\|^2_{Q_i} + \|u_i(k+j)\|^2_{R_i}] + \|x_i(k+N)\|^2_{P_i} \tag{48}$$

Note that the problem is separable in the cost function given by Eq. 47, while the coupling between the subproblems is due to the dynamics of Eq. 3. Define now the "coupling variables" $\nu_i = \sum_{j\neq i} A_{ij}x_j$ and write Eq. 3 as

$$x_i(k+1) = A_{ii}x_i(k) + B_iu_i(k) + \nu_i(k) \tag{49}$$

Let λ_i be the Lagrange multipliers, and consider the Lagrangian function:

$$\mathcal{L}(k) = \sum_{i=1}^{m} [J_i(k) + \sum_{l=0}^{N-1} \lambda_i(k+l)(\nu_i(k+l) - \sum_{j\neq i} A_{ij}x_j(k+l))] \tag{50}$$

For the generic vector variable φ, let $\bar{\varphi}_i(k) = [\varphi_i'(k)\,, \ldots\,, \varphi_i'(k+N-1)]'$ and $\bar{\varphi} = [\bar{\varphi}_1'\,, \ldots\,, \bar{\varphi}_m']'$. Then, by relaxation of the coupling constraints, the optimization problem of Eq. 47 can be stated as

$$\max_{\bar{\lambda}(k)} \min_{\bar{u}(k),\bar{\nu}(k)} \mathcal{L}(k) \tag{51}$$

or, equivalently

$$\max_{\bar{\lambda}(k)} \sum_{i=1}^{m} \tilde{J}_i(k) \tag{52}$$

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

$$\tilde{J}_i(k) = \min_{\bar{u}_i(k),\bar{\nu}_i(k)} [J_i(k) + \bar{\lambda}_i'(k)\bar{\nu}_i(k) - \sum_{j\neq i} \bar{\lambda}_j'(k)\bar{A}_{ji}\bar{x}_i(k))] \tag{53}$$

At any time instant, this optimization problem is solved according to the following two-step iterative procedure:

1) for a fixed $\bar{\lambda}$, solve the set of m independent minimization problems given by Eq. 53 with respect to $\bar{u}_i(k), \bar{\nu}_i(k)$;
2) given the collective values of $\bar{u}, \bar{\nu}$ computed at the previous step, solve the maximization problem given by Eq. 52 with respect to $\bar{\lambda}$.

Although the decomposition approaches usually require a great number of iterations to obtain a solution, many efforts have been devoted to derive efficient algorithms, see for example in [11], [112]. Notably, as shown for example in [35], the second step of the optimization procedure can be also performed in a distributed way by suitably exploiting the structure of the problem.

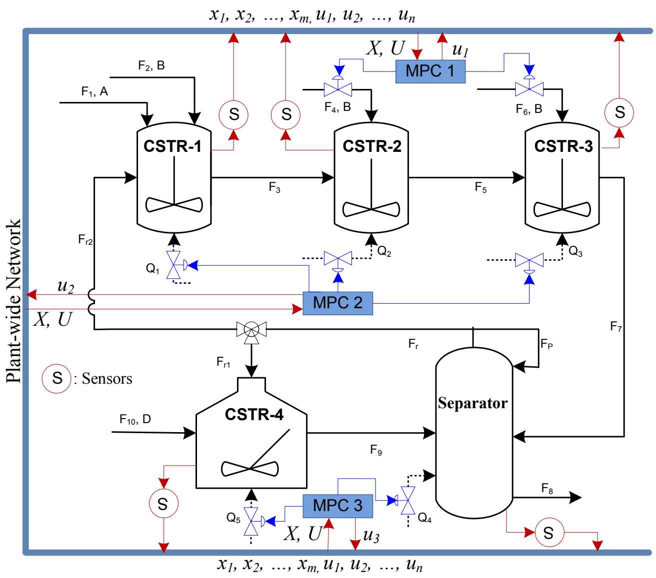

Fig. 11. A distributed MPC configuration for the alkylation of benzene with ethylene process.

D. Alkylation of benzene with ethylene process example (Cont')

Figure 11 shows a distributed control configuration for the alkylation process. In this design, three distributed MPCs are designed to manipulate the three different sets of control inputs and communicate through the plant-wide network to exchange information and coordinate their actions. Specifically, the first controller (MPC 1) is used to compute the values of Q_1, Q_2 and Q_3, the second distributed controller (MPC 2) is used to compute the values of Q_4 and Q_5, and the third controller (MPC 3) is used to compute the values of F_4 and F_6. This decomposition of the control loops is motivated by physical considerations: namely, one MPC (3) is used to manipulate the feed flow of ethylene into the process, another MPC (2) is used to manipulate the heat input/removal (Q_1, Q_2 and Q_3) to the first three reactors where the bulk of the alkylation reactions takes place and the third MPC (3) is used to manipulate the heat input/removal to the separator and the fourth reactor (Q_4 and Q_5) that processes the recycle stream from the separator. Either sequential or iterative communication architectures can be used in this DMPC design.

VI. Decompositions for DMPC

An important and unresolved in its generality issue in DMPC is how to decompose the total number of control actuators into small subsets, each one of them being controlled by a different MPC controller. There have been several ideas for how to do this decomposition based on plant layout considerations as well as via time-scale considerations. Below, we review some of these decompositions.

A. Decomposition into subsystems and multirate DMPC

Partitioning and decomposition of a process into several subsystems is an important topic. The recent work [60] describes the design of a network-based DMPC system using multirate sampling for large-scale nonlinear systems composed of several coupled subsystems. A schematic of the plant decomposition and of the control system is shown in Fig. 12. In the context of the alkylation of benzene with ethylene process example, this decomposition means that each reactor or separator has its own MPC controller, i.e., MPC 1 is used to manipulate Q_1, MPC 2 is used to manipulate Q_2 and F_4 and so on. Specifically, in [60], the states of each local subsystem are assumed to be divided into fast sampled states and slowly sampled states. Furthermore, the assumption is made that there is a distributed controller associated with each subsystem and the distributed controllers are connected through a shared communication network. At a sampling time in which slowly and fast sampled states are available, the distributed controllers coordinate their actions and predict future input trajectories which, if applied until the next instant that both slowly and fast sampled states are available, guarantee closed-loop stability. At a sampling time in which only fast sampled states are available, each distributed controller tries to further optimize the input trajectories calculated at the last instant in which the controllers communicated, within a constrained set of values to improve the closed-loop performance with the help of the available fast sampled states of its subsystem.

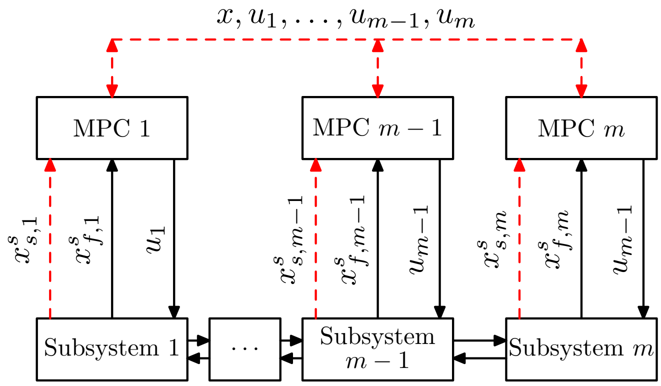

Fig. 12. Multirate DMPC architecture (solid line denotes fast state sampling and/or point-to-point links; dashed line denotes slow state sampling and/or shared communication networks).

B. Hierarchical and multilevel MPC

In the process industry, the control structure is usually organized in a number of different layers. At the bottom level, standard PI-PID regulators are used for control of the actuators, while at a higher layer MPC is usually applied for set-point tracking of the main control variables. Finally, at the top of the hierarchy, optimization is used for plantwide control with the scope of providing efficient, cost-effective, reliable, and smooth operation of the entire plant. An extensive discussion of hierarchical, multilayer control is beyond the scope of this review, and reference is made to the wide literature in the field, with particular reference to the excellent and recent survey papers [149], [40]. Recent results on the design of two-level control systems designed

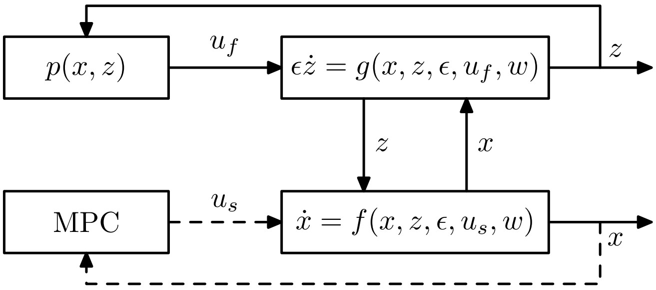

Fig. 13. A schematic of a composite control system using MPC in the slow time-scale.

with MPC and allowing for reconfiguration of the control structure have also been reported in [122], [155]. As an additional remark, it is worth mentioning that a recent stream of research is devoted to the so-called economic MPC, with the aim to directly use feedback control for optimizing economic performance, rather than simply stabilizing the plant and maintaining steady operation, see e.g. [127], [34], [58].

In a wider perspective, hierarchical and multilayer structures are useful for control of very large scale systems composed by a number of autonomous or semi-autonomous subsystems, which must be coordinated to achieve a common goal. Examples can be found in many different fields, such as robotics [8], [152], transportation networks [114], voltage control in energy distribution networks [115], control of irrigation canals [116], [161], and automation of baggage handling systems [147]. The design of multilayer structures according to a leader-follower approach for networked control has been considered in [7]. In any case, the design of multilayer structures requires multi-level and multi-resolution models, which, according to [149], can be obtained according to a functional, temporal or spatial decomposition approach.

C. MPC of Two-Time-Scale Systems

Most chemical processes involve physico-chemical phenomena that occur in distinct (slow and fast) time scales. Singular perturbation theory provides a natural framework for modeling, analyzing and controlling multiple time-scale processes. While there has been extensive work on feedback control of two-time-scale processes within the singular perturbation framework (e.g., [72]), results on MPC of two-time-scale systems have been relatively recent [153], [19]. Below, we discuss some of these results pertaining to the subject of decentralized/distributed MPC.

1) Slow time-scale MPC: Specifically, in [19], MPC was considered in the context of nonlinear singularly perturbed systems in standard form with the following state-space description:

$$\begin{array}{rclcrcl} \dot{x} & = & f(x,z,\epsilon,u_s,w), & & x(0) & = & x_0 \\ \epsilon\dot{z} & = & g(x,z,\epsilon,u_f,w), & & z(0) & = & z_0 \end{array} \tag{54}$$

where $x \in R^n$ and $z \in R^m$ denote the vector of state variables, ϵ is a small positive parameter, $w \in R^l$ denotes the vector of disturbances and $u_s \in U \subset R^p$ and $u_f \in \tilde{V} \subset R^q$ are two sets of manipulated inputs. Since the small parameter ϵ multiplies the time derivative of the vector z in the system of Eq. 2, the separation of the slow and fast variables in Eq. 2 is explicit, and thus, we will refer to the vector x as the slow states and to the vector z as the fast states. With respect to the control problem formulation, the assumption is made that the fast states z are sampled continuously and their measurements are available for all time t (for example, variables for which fast sampling is possible usually include temperature, pressure and hold-ups) while the slow states x are sampled synchronously and are available at time instants indicated by the time sequence $\{t_{k\geq 0}\}$ with $t_k = t_0 + k\Delta,\ k = 0,1,\ldots$ where t_0 is the initial time and Δ is the sampling time (for example, slowly sampled variables usually involve species concentrations). The set of manipulated inputs u_f is responsible for stabilizing the fast dynamics of Eq. 2 and for this set the control action is assumed to be computed continuously, while the set of manipulated inputs u_s is evaluated at each sampling time t_k and is responsible for stabilizing the slow dynamics and enforcing a desired level of optimal closed-loop performance. The explicit separation of the slow and fast variables in the system of Eq. 2 allows decomposing it into two separate reduced-order systems evolving in different time-scales. To proceed with such a two-time-scale decomposition and in order to simplify the notation of the subsequent development, we will first address the issue of stability of the fast dynamics. Since there is no assumption that the fast dynamics of Eq. 2 are asymptotically stable, we assume the existence of a "fast" feedback control law $u_f = p(x,z)$ that renders the fast dynamics asymptotically stable. Substituting $u_f = p(x,z)$ in Eq. 2 and setting $\epsilon = 0$ in the resulting system, we obtain:

$$\frac{dx}{dt} = f(x,z,0,u_s,w) \tag{55a}$$

$$0 = g(x,z,0,p(x,z),w) \tag{55b}$$

Assuming that the equation $g(x,z,0,p(x,z),w) = 0$ possesses a unique root

$$z = \hat{g}(x,w) \tag{56}$$

we can construct the slow subsystem:

$$\frac{dx}{dt} = f(x,\hat{g}(x,w),0,u_s,w) =: f_s(x,u_s,w) \tag{57}$$

Introducing the fast time scale $\tau = \dfrac{t}{\epsilon}$ and the deviation variable $y = z - \hat{g}(x,w)$, we can rewrite the nonlinear singularly perturbed system of Eq. 2 as follows:

$$\begin{array}{rcl} \dfrac{dx}{d\tau} & = & \epsilon f(x,y+\hat{g}(x,w),\epsilon,u_s,w) \\ \dfrac{dy}{d\tau} & = & g(x,y+\hat{g}(x,w),\epsilon,u_f,w) - \epsilon\dfrac{\partial\hat{g}}{\partial w}\dot{w} \\ & & -\epsilon\dfrac{\partial\hat{g}}{\partial x}f(x,y+\hat{g}(x,w),\epsilon,u_s,w) \end{array} \tag{58}$$

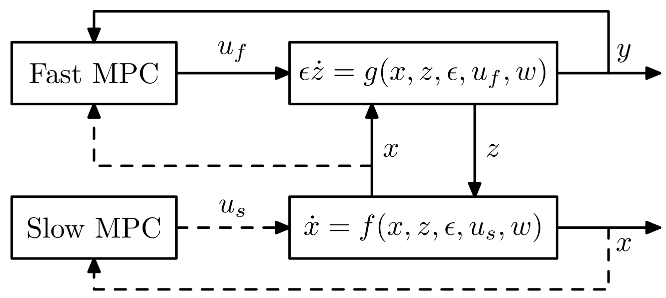

Fig. 14. A schematic of a composite control system using MPC in both the fast and slow time-scales.

Setting $\epsilon = 0$, we obtain the following fast subsystem:

$$\frac{dy}{d\tau} = g(x, y + \hat{g}(x, w), 0, u_f, w) \tag{59}$$

where x and w can be considered as "frozen" to their initial values. The fast subsystem can be made asymptotically stable uniformly in $x \in R^n$ and $w \in R^l$ with the appropriate design of $u_f = p(x, z)$. MPC is used to compute the control action u_s in the slow time-scale. A schematic of the proposed control system structure is shown in Fig. 13. Specifically, an LMPC of the type of Eq. 16 was used in [103] which guarantees practical stability of the closed-loop system and allows for an explicit characterization of the stability region to compute u_s. The LMPC is based on the Lyapunov-based controller $h(x)$. Using stability results for nonlinear singularly perturbed systems, the closed-loop system is analyzed and sufficient conditions for stability have been derived [19].

2) Fast/Slow MPC design: In addition to the development of the composite control system of Fig. 13, the singular perturbation framework of Eq. 54 can be also used to develop composite control systems where an MPC controller is used tin the fast time scale. In this case, a convenient way from a control problem formulation point of view is to design a fast-MPC that uses feedback of the deviation variable y in which case u_f is only active in the boundary layer (fast motion of the fast dynamics) and becomes nearly zero in the slow time-scale. The resulting control architecture in this case is shown in Figure 14 where there is no need for communication between the fast MPC and the slow MPC; in this sense, this control structure can be classified as decentralized. Specifically, referring to the singularly perturbed system of Eq. 58, the cost can be defined as

$$\begin{aligned} J &= J_s + J_f \\ &= \int_0^{N\Delta_s} \left[x^T(\tilde{\tau}) Q_s x(\tilde{\tau}) + u_s^T(\tilde{\tau}) R_s u_s(\tilde{\tau})\right] d\tilde{\tau} \\ &\quad + \int_0^{N\Delta_f} \left[y^T(\tilde{\tau}) Q_f y(\tilde{\tau}) + u_f^T(\tilde{\tau}) R_f u_f(\tilde{\tau})\right] d\tilde{\tau} \end{aligned} \tag{60}$$

where Q_s, Q_f, R_s, R_f are positive definite weighting matrices, Δ_s is the sampling time of u_s and Δ_f is the sampling time of u_f. The fast MPC can be then formulated as follows

$$\min_{u_f \in S(\Delta_f)} J_f \tag{61a}$$

$$\text{s.t. } \frac{dy}{d\tau} = g(x, y + \tilde{g}(x, 0), 0, u_f, 0) \tag{61b}$$

$$u_f \in \bar{V} \tag{61c}$$

$$\text{stability constraints} \tag{61d}$$

where $z = \tilde{g}(x, 0)$ is the solution of the equation $g(x, z, 0, 0, 0) = 0$. The slow MPC is designed on the basis of the system of Eq. 57 with $w = 0$ and $\hat{g}(x, w) = \tilde{g}(x)$. Such a two-time-scale DMPC architecture takes advantage of the time-scale separation in the process model and does not require communication between the two MPCs yet can ensure closed-loop stability and near optimal performance in the sense of computing control actions that minimize $J = J_s + J_f$ as $\epsilon \to 0$.

VII. Distributed state estimation and asynchronous/delayed sampling

A. Distributed state estimation

Many algorithms for distributed state estimation have already been proposed in the literature. Among them, we can recall the early contributions reported in [57], [126], aimed at reducing the computational complexity of centralized Kalman filters. In [108], a solution based on reduced-order and decoupled models for each subsystem was proposed, while subsystems with overlapping states were considered in the fully distributed schemes of [70], [142], [141] and in [151], where an all-to-all communication among subsystems was required. The problem of distributed estimation for sensor networks where each sensor measures just some of the system outputs and computes the estimate of the overall state have been studied in [5], [17], [66], [120] and in [41], [42], [43], where the Moving Horizon Estimation (MHE) approach has been used to cope with constraints on noise and state variables. Finally, MHE algorithms for linear constrained systems decomposed into interconnected subsystems without overlapping states have been described in [44], [45].

B. Asynchronous and Delayed Feedback

Previous work on MPC design for systems subject to asynchronous or delayed feedback has primarily focused on centralized MPC designs [16], [133], [135], [63], [79], [107], [84], [54], [123], [48]. In a recent work [51], the issue of delays in the communication between distributed controllers was addressed. In addition to these works, control and monitoring of complex distributed systems with distributed intelligent agents were studied in [148], [27], [121].

Below, we review a recent iterative DMPC scheme [83], [81], taking into account asynchronous and delayed measurements explicitly in its formulation and providing deterministic closed-loop stability properties.

C. Iterative DMPC with Asynchronous, Delayed Feedback

We assume that feedback of the state of the system of Eq. 2, $x(t)$, is available at asynchronous time instants t_a where $\{t_{a\geq 0}\}$ is a random increasing sequence of times; that is, the intervals between two consecutive instants are not fixed. The distribution of $\{t_{a\geq 0}\}$ characterizes the time the feedback loop is closed or the time needed to obtain a new state measurement. In general, if there exists the possibility of arbitrarily large periods of time in which feedback is not available, then it is not possible to provide guaranteed stability properties, because there exists a non-zero probability that the system operates in open-loop for a period of time large enough for the state to leave the stability region. In order to study the stability properties in a deterministic framework, we assume that there exists an upper bound T_m on the interval between two successive time instants in which the feedback loop is closed or new state measurements are available, that is:

$$\max_{a}\{t_{a+1}-t_a\} \leq T_m. \tag{62}$$

Furthermore, we also assume that there are delays in the measurements received by the controllers due to delays in the sampling process and data transmission. In order to model delays in measurements, another auxiliary variable d_a is introduced to indicate the delay corresponding to the measurement received at time t_a, that is, at time t_a, the measurement $x(t_a-d_a)$ is received. In order to study the stability properties in a deterministic framework, we assume that the delays associated with the measurements are smaller than an upper bound D. Both assumptions are reasonable from process control and networked control systems perspectives [156], [157], [111], [104] and allow us to study deterministic notions of stability. This model of feedback/measurements is of relevance to systems subject to asynchronous/delayed measurement samplings and to networked control systems, where the asynchronous/delayed feedback is introduced by data losses/traffic in the communication network connecting the sensors/actuators and the controllers.

In the presence of asynchronous/delayed measurements, the iterative DMPC presented in Section V-B.2 cannot guarantee closed-loop stability and both the implementation strategy and the formulation of the distributed controllers has to take into account the occurrence of asynchronous/delayed measurements. Specifically, we take advantage of the system model both to estimate the current system state from a delayed measurement and to control the system in open-loop when new information is not available. To this end, when a delayed measurement is received, the distributed controllers use the system model and the input trajectories that have been applied to the system to get an estimate of the current state and then based on the estimate, MPC optimization problems are solved to compute the optimal future input trajectory that will be applied until new measurements are received. The implementation strategy for the iterative DMPC design is as follows:

1. When a measurement $x(t_a-d_a)$ is available at t_a, all the distributed controllers receive the state measurement and check whether the measurement provides new information. If $t_a-d_a > \max_{l<a} t_l - d_l$, go to Step 2. Else the measurement does not contain new information and is discarded, go to Step 3.
2. All the distributed controllers estimate the current state of the system $x^e(t_a)$ and then evaluate their future input trajectories in an iterative fashion with initial input guesses generated by $h(\cdot)$.
3. At iteration c ($c \geq 1$):
   3.1. Each controller evaluates its own future input trajectory based on $x^e(t_a)$ and the latest received input trajectories of all the other distributed controllers (when $c = 1$, initial input guesses generated by $h(\cdot)$ are used).
   3.2. The controllers exchange their future input trajectories. Based on all the input trajectories, each controller calculates and stores the value of the cost function.
4. If a termination condition is satisfied, each controller sends its entire future input trajectory corresponding to the smallest value of the cost function to its actuators; if the termination condition is not satisfied, go to Step 3 ($c \leftarrow c+1$).

1) When a new measurement is received ($a \leftarrow a+1$), go to Step 1.

In order to estimate the current system state $x^e(t_a)$ based on a delayed measurement $x(t_a-d_a)$, the distributed controllers take advantage of the input trajectories that have been applied to the system from t_a-d_a to t_a and the system model of Eq. 2. Let us denote the input trajectories that have been applied to the system as $u^*_{d,i}(t)$, $i=1,\dots,m$. Therefore, $x^e(t_a)$ is evaluated by integrating the following equation:

$$\dot{x}^e(t) = f(x^e(t)) + \sum_{i=1}^{m} g_i(x^e(t))u^*_{d,i}(t), \forall t \in [t_a-d_a, t_a) \tag{63}$$

with $x^e(t_a-d_a) = x(t_a-d_a)$.

Before going to the design of the iterative DMPC, we need to define another nominal sampled trajectory $\check{x}(t|t_a)$ for $t \in [t_a, t_a+N\Delta)$, which is obtained by replacing $\hat{x}(t|t_a)$ with $\check{x}(t|t_a)$ in Eq. 38 and then integrating the equation with $\check{x}(t_a|t_a) = x^e(t_a)$. Based on $\check{x}(t|t_a)$, we define a new input trajectory as follows:

$$\begin{aligned} u^e_{n,j}(t|t_a) &= h_j(\check{x}(t_a+l\Delta|t_a)), j=1,\dots,m, \\ &\forall t \in [t_a+l\Delta, t_a+(l+1)\Delta), l=0,\dots,N-1 \end{aligned} \tag{64}$$

which will be used in the design of the LMPC to construct the stability constraint and used as the initial input guess for iteration 1 (i.e., $u^{*,0}_{d,i} = u^e_{n,i}$ for $i=1,\dots,m$).

Specifically, the design of LMPC j, $j=1,\dots,m$, at iteration c is based on the following optimization problem:

$$\min_{u_j \in S(\Delta)} J(t_a) \tag{65a}$$

$$\text{s.t.}\ \dot{\tilde{x}}^j(t) = f(\tilde{x}^j(t)) + \sum_{i=1}^{m} g_i(\tilde{x}^j(t))u_i(t) \tag{65b}$$

$$u_i(t) = u_{d,i}^{*,c-1}(t|t_a), \forall i \neq j \quad (65c)$$

$$\left\| u_j(t) - u_{d,j}^{*,c-1}(t|t_a) \right\| \leq \Delta u_j,$$
$$\forall \tau \in [t_a, t_a + N_{D,a}\Delta) \quad (65d)$$

$$u_j(t) \in U_j \quad (65e)$$

$$\tilde{x}^j(t_a) = x^e(t_a) \quad (65f)$$

$$\frac{\partial V(\tilde{x}^j(t))}{\partial \tilde{x}^j} \left(\frac{1}{m} f(\tilde{x}^j(t)) + g_j(\tilde{x}^j(t)) u_j(t) \right)$$
$$\leq \frac{\partial V(\check{x}(t|t_a))}{\partial \check{x}} \left(\frac{1}{m} f(\check{x}(t|t_a)) + g_j(\check{x}(t|t_a)) u_{n,j}^e(t|t_a) \right),$$
$$\forall t \in [t_a, t_a + N_{D,a}\Delta) \quad (65g)$$

where $N_{D,a}$ is the smallest integer satisfying $N_{D,a}\Delta \geq T_m + D - d_a$. The optimal solution to this optimization problem is denoted $u_{d,j}^{*,c}(a|t_a)$ which is defined for $t \in [t_a, t_a + N\Delta)$. Accordingly, we define the final optimal input trajectory of LMPC j of Eq. 65 as $u_{d,j}^*(t|t_k)$ which is also defined for $t \in [t_a, t_a + N\Delta)$. Note again that the length of the constraint $N_{D,a}$ depends on the current delay d_a, so it may have different values at different time instants and has to be updated before solving the optimization problems.

The manipulated inputs of the closed-loop system under the above iterative DMPC for systems subject to delayed measurements are defined as follows:

$$u_i(t) = u_{d,i}^*(t|t_a), i = 1, \ldots, m, \forall t \in [t_a, t_{a+q}) \quad (66)$$

for all t_a such that $t_a - d_a > \max_{l<a} t_l - d_l$ and for a given t_a, the variable q denotes the smallest integer that satisfies $t_{a+q} - d_{a+q} > t_a - d_a$. Recent work has also addressed the problem of communication disruptions between the distributed controllers [59].

VIII. Future Research Directions

In this section, we discuss various topics for future research work in the area of DMPC; the list is not intended to be exhaustive and it is certainly based on our experiences, biases and hopes.

A. DMPC: Loop Partitioning and Decompositions

While there have been several suggestions for how to partition the loops in a DMPC system (i.e., what specific control actuators each MPC will manipulate) based on physical arguments, insight into process dynamic behavior like, for example, two-time-scale behavior [74], [65], or plant layout considerations like one controller per plant unit, there is no general framework for computing optimal (in a certain well-defined sense) input (control actuator) decompositions for DMPC. Undoubtedly, this problem is very hard in its full generality, however, even solutions for large-scale systems of specific structure, like linear systems [106], [1] or well-defined parts of a chemical plant flowsheet, could be very useful. Research in this direction should go hand-in-hand with the development of optimal communication strategies between the distributed controllers so that controller evaluation time, communication network usage and closed-loop stability, performance and robustness are optimized.

B. Distributed State Estimation and DMPC

Most of the available DMPC schemes rely on the assumption of availability of measurements of the complete state vector. However, it is possible that a distributed controller in a large-scale control system may not have access to all the measurements or that measurements of all the process states are not available. In this case, in the design of the distributed controllers, we need to take into account that different distributed controllers may have access to measurements of different parts of the process states, so that methods for DMPC with partial state measurements are required. Future research in this direction should take advantage of the available distributed state estimation schemes reviewed in section VII of this paper and should look at how best the combination of a DMPC algorithm with centralized/distributed state estimators can be addressed. One approach is to design a different state observer for each controller (i.e., distributed state estimation), while an alternative approach is to design a centralized observer that sends the estimated state to all the distributed controllers. In this context, the integration of the state estimation schemes with the DMPC algorithms so that desired levels of stability, performance and robustness are attained in the closed-loop system should be rigorously studied.

C. Economic DMPC and distributed optimization

Most industrial process control applications are based on hierarchical control schemes in which first the operation point of the plant is determined based on economic, safety and environmental considerations (usually using steady state models), and then process control systems are used to drive the plant to this steady state (usually using dynamic models). There have been some recent advances in integrating economic optimization and process control in a single MPC using centralized approaches, for example the MPC for tracking schemes proposed in [30], [47], [77] and the MPC schemes based directly on an economic cost function proposed in [34], [62], [58]. It is clear that, for large scale systems, DMPC may be an appropriate path to tackle the resulting economic optimization problem. Furthermore, DMPC stands to benefit from distributed optimization schemes that are tailored to handle DMPC optimization problems in an optimal fashion accounting for control-loop decomposition, plant variable interaction patterns and controller communication strategies. Research in this direction may start from the extension to the distributed case of well-known techniques for centralized MPC, such as the multiple shooting method.

D. DMPC and Hybrid systems

Hybrid systems constitute an important class of mathematical models that explicitly account for the intricate coupling between continuous dynamics and discrete events. While there has been extensive work over the last fifteen years on analysis and control of hybrid systems (see, for example, [10], [25] and the references therein), distributed MPC of hybrid systems is a research topic that has received no attention. In the context of chemical process control and operations, due to changes in raw materials, energy sources, product specifications and market demands, and abrupt actuator and sensor faults, it is possible to describe process behavior with classes of switched nonlinear systems that involve differential equation models whose right-hand-side is indexed with respect to different modes of operation. From a controller design standpoint, in order to achieve closed-loop stability, discrete mode transition situations should be carefully accounted for in the control problem formulation and solution. In order to achieve mode transitions in an optimal setting and accommodate input/state constraints, distributed model predictive control (MPC) framework can be employed, particularly in cases where the computational complexity of a centralized MPC may significantly increase as the number of operational modes, control inputs and states increases.

E. Monitoring and reconfigurability of DMPC

Monitoring and reconfiguration of DMPC is an important research topic. DMPC systems offer a vast set of possibilities for reconfiguration in the event of sensor and actuator faults to maintain the desired closed-loop performance. In a recent set of papers [23], [22], a data-based monitoring and reconfiguration system was developed for a distributed model predictive control system in the presence of control actuator faults. In addition to a monitoring method, appropriate DMPC reconfiguration (fault-tolerant control) strategies were designed to handle the actuator faults and maintain the closed-loop system state within a desired operating region. There is certainly a lot more to be done in the context of DMPC monitoring and fault-tolerance.

Furthermore, in addition to its importance in the context of DMPC fault-tolerance, reconfigurability of DMPC could provide flexibility to the control system and could be explored in the context of other areas as follows.

During steady-state operation, it is not necessary to continuously transmit among the distributed estimation/control agents. In fact, in the case where one system does not receive any new information (and can be sure that no transmission faults have occurred), it can be assumed that the other agents basically maintain their previous state. This reduction of the information transmitted can be particularly significant in sensor networks with local power supply in order to have significant energy savings, which could guarantee a longer "life" of the sensors and/or of the actuators.

For similar reasons, future research efforts could deal with the so-called "plug and play" control as well as with DMPC scalability. In nominal operating conditions, the control system assumes a minimal configuration, while in perturbed conditions sensors and/or actuators are added. The challenge here is to avoid the redesign of the overall control system, in particular for the elements not directly dynamically connected with the additional sensors and actuators. The reader may refer to [105], [145], [71], [155] for some references on "plug and play" control.

In addition, moving from industrial plants to very large-scale systems, such as transportation or distribution networks, or to the so-called "System-of-Systems" (i.e., very large-scale infrastructures of interacting subsystems, which are by themselves composed of large-scale and complex systems; see, for example, [93]), with operational and managerial independence, it is clear that the problem of reconfiguration of the control system is fundamental to cope with changing requirements. For these systems, also the problem of partitioning and clustering is a very important one (recent work can be found in [118]). In general, there is a substantial lack of methodologies for appropriate temporal and spatial partitions and for the development of consistent multi-level, multi-scale models for DMPC design.

F. Applications

DMPC has a lot to offer in the context of industrial process control practice. As plants become increasingly automated with advanced model-based control systems and the adoption of advanced communication networks together with the associated sensors and actuators continuous to broaden, DMPC could provide the framework for the design of the next-generation, distributed model-based control systems. But the impact of DMPC could go well-beyond industrial process control practice and could become the method of choice for the design of control systems for the individual componenets/subsystems of large-scale, heterogenous distributed networks (like, for example, "smart grid"-type networks where numerous renewables-based energy generation systems are coupled with the electric grid and "smart" loads). It is our hope that this paper will contribute towards developing further DMPC theory and practice.

References

[1] W. Al-Gherwi, H. Budman, and A. Elkamel. Selection of control structure for distributed model predictive control in the presence of model errors. *Journal of Process Control*, 20:270–284, 2010.

[2] A. Alessio, D. Barcelli, and A. Bemporad. Decentralized model predictive control of dynamically coupled linear systems. *Journal of Process Control*, 21:705–714, 2011.

[3] A. Alessio and A. Bemporad. Stability conditions for decentralized model predictive control under packet drop communication. In *Proceedings of the American Control Conference*, pages 3577–3582, Seattle, Washington, 2008.

[4] F. Allgöwer and F. J. Doyle. Nonlinear process control - which way to the promised land? In *Proceedings of 5th International Conference on Chemical Process Control*, pages 24–45, Tahoe City, CA, 1997.

[5] P. Alriksson and A. Rantzer. Distributed Kalman filtering using weighted averaging. In *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, Kyoto, Japan, 2006.

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

[7] T. Baçar and R. Srikant. A Stackelberg network game with a large number of followers. *Journal of Optimization, Theory and Applications*, 115:479–490, 2002.

[8] A. D. Baker. A survey of factory control algorithms that can be implemented in a multi-agent heterarchy: Dispatching, scheduling, and pull. *Journal of Manufacturing Systems*, 17:297–320, 1998.

[9] L. Bakule. Decentralized control: An overview. *Annual Reviews in Control*, 32:87–98, 2008.

[10] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics and constraints. *Automatica*, 35:407–427, 1999.

[11] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation*. Athena Scinetific, Belmont, Massachusetts, 1997.

[12] R. R. Bitmead, M. Gevers, and V. Wertz. *Adaptive Optimal Control – The Thinking Man's GPC*. Prentice-Hall, Englewood Cliffs, NJ, 1990.

[13] E.F. Camacho and C. Bordons. *Model Predictive Control*. Springer Verlag, 2004.

[14] E. Camponogara and B. Barcelos de Oliveira. Distributed optimization for model predictive control of linear-dynamic networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 39:1331–1338, 2009.

[15] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, 22:44–52, 2002.

[16] P. Carini, R. Micheli, and R. Scattolini. Multirate self-tuning predictive control with application to a binary distillation column. *International Journal of Systems Science*, 21:51–64, 1990.

[17] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri. Distributed Kalman filtering based on consensus strategies. *IEEE Journal on Selected Areas in Communications*, 26:622–633, 2008.

[18] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34:1205–1217, 1998.

[19] X. Chen, M. Heidarinejad, J. Liu, D. Muñoz de la Peña, and P. D. Christofides. Model predictive control of nonlinear singularly perturbed systems: Application to a large-scale process network. *Journal of Process Control*, 21:1296–1305, 2011.

[20] R. Cheng, J. F. Forbes, and W.S. Yip. Price-driven coordination methods for solving plant-wide MPC problems. *Journal of Process Control*, 17:429–438, 2007.

[21] R. Cheng, J. F. Forbes, and W.S. Yip. Dantzig-Wolfe decomposition and plant-wide MPC coordination. *Computers and Chemical Engineering*, 32:1507–1522, 2008.

[22] D. Chilin, J. Liu, J. F. Davis, and P. D. Christofides. Data-based monitoring and reconfiguration of a distributed model predictive control system. *International Journal of Robust and Nonlinear Control*, in press.

[23] D. Chilin, J. Liu, D. Muñoz de la Peña, P. D. Christofides, and J. F. Davis. Detection, isolation and handling of actuator faults in distributed model predictive control systems. *Journal of Process Control*, 20:1059–1075, 2010.

[24] P. D. Christofides, J. F. Davis, N. H. El-Farra, D. Clark, K. R. D. Harris, and J. N. Gipson. Smart plant operations: Vision, progress and challenges. *AIChE Journal*, 53:2734–2741, 2007.

[25] P. D. Christofides and N. H. El-Farra. *Control of nonlinear and hybrid process systems: Designs for uncertainty, constraints and time-delays*. Springer-Verlag, Berlin, Germany, 2005.

[26] P. D. Christofides, J. Liu, and D. Muñoz de la Peña. *Networked and Distributed Predictive Control: Methods and Nonlinear Process Network Applications*. Advances in Induatrial Control Series. Springer-Verlag, London, England, 2011.

[27] A. Cinar, A. Palazoglu, and F. Kayihan. *Chemical Process Performance Evaluation*. CRC Press-Taylor & Francis, Boca Raton, FL, 2007.

[28] F. Clarke, Y. Ledyaev, and E. Sontag. Asymtotic controllability implies feedback stabilization. *IEEE Transactions on Automatic Control*, 42:1394–1407, 1997.

[29] H. Cui and E. W. Jacobsen. Performance limitations in decentralized control. *Journal of Process Control*, 12:485–494, 2002.

[30] Limon D., I. Alvarado, T. Alamo, and E.F. Camacho. MPC for tracking piecewise constant references for constrained linear systems. *Automatica*, 44:2382 – 2387, 2008.

[31] J. F. Davis. *Report from NSF Workshop on Cyberinfrastructure in Chemical and Biological Systems: Impact and Directions,* (see http://www.oit.ucla.edu/nsfci/NSFCIFullReport.pdf for the pdf file of this report). Technical report, 2007.

[32] E. J. Davison and T. N. Chang. Decentralized stabilization and pole assisgnment for general proper systems. *IEEE Transactions on Automatic Control*, 35:652–664, 1990.

[33] G. De Nicolao, L. Magni, and R. Scattolini. Stabilizing receding-horizon control of nonlinear time-varying systems. *IEEE Trans. on Automatic Control*, AC-43:1030–1036, 1998.

[34] M. Diehl, R. Amrit, and J. B. Rawlings. A Lyapunov function for economic optimizing model predictive control. *IEEE Transactions on Automatic Control*, 56:703–707, 2011.

[35] M.D. Doan, T. Keviczky, and B. De Schutter. An iterative scheme for distributed model predictive control using Fenchel's duality. *Journal of Process Control*, 21:746–755, 2011.

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

[37] W. B. Dunbar and S. Desa. Distributed MPC for dynamic supply chain management. In *Int. Workshop on Assessment and Future Directions of NMPC, Freudenstadt-Lauterbad, Germany, 26-30*, 2005.

[38] W. B. Dunbar and R. M. Murray. Distributed receding horizon control with application to multi-vehicle formation stabilization. *Automatica*, 42:549–558, 2006.

[39] Camponogara E., H. F. Scherer, and L. Vila Moura. Distributed optimization for predictive control with input and state constraints: Preliminary theory and application to urban traffic control. In *Proceedings of the IEEE Int. Conf. on Systems, Man, and Cybernetics*, pages 3726–3732, 2009.

[40] S. Engell. Feedback control for optimal process operation. *Journal of Process Control*, 17:203–219, 2007.

[41] M. Farina, G. Ferrari-Trecate, and R. Scattolini. Distributed moving horizon estimation for sensor networks. In *Proceedings of the 1st IFAC Workshop on Estimation and Control of Networked Systems, Venice, Italy*, pages 126 – 131, 2009.

[42] M. Farina, G. Ferrari-Trecate, and R. Scattolini. Distributed moving horizon estimation for linear constrained systems. *IEEE Transactions on Automatic Control*, 55:2462–2475, 2010.

[43] M. Farina, G. Ferrari-Trecate, and R. Scattolini. Distributed moving horizon estimation for nonlinear constrained systems. In *Proceedings of the 8th IFAC Symposium on Nonlinear Control Systems*, Bologna, Italy, 2010.

[44] M. Farina, G. Ferrari-Trecate, and R. Scattolini. Moving horizon state estimation of large-scale constrained partitioned systems. *Automatica*, 46:910 – 918, 2010.

[45] M. Farina, G. Ferrari-Trecate, and R. Scattolini. Moving horizon estimation for distributed nonlinear systems with application to cascade river reaches. *Journal of Process Control*, 21:767–774, 2011.

[46] M. Farina and R. Scattolini. Distributed non-cooperative MPC with neighbor-to-neighbor communication. In *Proceedings of the 18th IFAC World Congress*, Milan, Italy, 2011.

[47] A. Ferramosca, D. Limon, I. Alvarado, T. Alamo, and E.F. Camacho. MPC for tracking with optimal closed-loop performance. *Automatica*, 45:1975 – 1978, 2009.

[48] R. Findeisen and P. Varutti. Stabilizing nonlinear predictive control over nondeterministic communication networks. In Lalo Magni, Davide Raimondo, and Frank Allgwer, editors, *Nonlinear Model Predictive Control*, volume 384 of *Lecture Notes in Control and Information Sciences*, pages 167–179. Springer Berlin / Heidelberg, 2009.

[49] W. Findeisen, F.N. Bailey, M. Brdys, K. Malinovski, P. Tatjewski, and A. Wozniak. *Control and Coordination in Hierarchical Systems*. Wiley and Sons, Ne York, 1980.

[50] F. A. Fontes. A general framework to design stabilizing nonlinear model predictive controllers. *Systems & Control Letters*, 42:127–143, 2001.

[51] E. Franco, L. Magni, T. Parisini, M. M. Polycarpou, and D. M. Raimondo. Cooperative constrained control of distributed agents with nonlinear dynamics and delayed information exchange: A stabilizing receding-horizon approach. *IEEE Transactions on Automatic Control*, 53:324–338, 2008.

[52] C. E. García, D. M. Prett, and M. Morari. Model predictive control: Theory and practice–A survey. *Automatica*, 25:335–348, 1989.

[53] G. Grimm, M. J. Messina, S. E. Tuna, and A. R. Teel. Model predictive control: For want of a local control Lyapunov function, all is not lost. *IEEE Transactions on Automatic Control*, 50:546–558, 2005.

[54] L. Grüne, J. Pannek, and K. Worthmann. A networked unconstrained nonlinear MPC scheme. In *Proceedings of European Control Conference*, pages 371–376, Budapest, Hungary, 2009.

[55] R. D. Gudi and J. B. Rawlings. Identification for decentralized model predictive control. *AIChE Journal*, 52:2198–2210, 2006.

[56] E. Gyurkovics and A. M. Elaiw. Stabilization of sampled-data nonlinear systems by receding horizon contyrol via discrete-time approximations. *Automatica*, 40:2017–2028, 2004.

[57] H.R. Hashemipour, S. Roy, and A.J. Laub. Decentralized structures

for parallel Kalman filtering. *IEEE Transactions on Automatic Control*, 33:88 – 94, 1988.

[58] M. Heidarinejad, J. Liu, and P. D. Christofides. Economic model predictive control of nonlinear process systems using Lyapunov techniques. *AIChE Journal*, in press.

[59] M. Heidarinejad, J. Liu, D. Muñoz de la Peña, J. F. Davis, and P. D. Christofides. Handling communication disruptions in distributed model predictive control of nonlinear systems. *Journal of Process Control*, 21:173–181, 2011.

[60] M. Heidarinejad, J. Liu, D. Muñoz de la Peña, J. F. Davis, and P. D. Christofides. Multirate Lyapunov-based distributed model predictive control of nonlinear uncertain systems. *Journal of Process Control*, 21:1231–1242, 2011.

[61] M. A. Henson and D. E. Seborg. *Nonlinear Process Control*. Prentice-Hall, Englewood Cliffs, NJ, 1997.

[62] R. Huang, E. Harinath, and L. T. Biegler. Lyapunov stability of economically-oriented nmpc for cyclic processes. *Journal of Process Control*, 21:501–509, 2011.

[63] S. C. Jeong and P. Park. Constrained MPC algorithm for uncertain time-varying systems with state-delay. *IEEE Transactions on Automatic Control*, 50:257–263, 2005.

[64] D. Jia and B. H. Krogh. Distributed model predictive control. In *Proceedings of the American Control Conference*, pages 2767–2772, Arlington, VA, USA, 2001.

[65] S. S. Jogwar, M. Baldea, and P. Daoutidis. Dynamics and control of process networks with large energy recycle. *Industrial & Engineering Chemistry Research*, 48:6087–6097, 2009.

[66] M. Kamgarpour and C. Tomlin. Convergence properties of a decentralized Kalman filter. *Proceedings of the* 47^{th} *IEEE Conference on Decision and Control*, pages 3205 – 3210, 2008.

[67] M. R. Katebi and M. A. Johnson. Predictive control design for large-scale systems. *Automatica*, 33:421–425, 1997.

[68] T. Keviczky, F. Borrelli, and G. J. Balas. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, 42:2105–2115, 2006.

[69] T. Keviczky, F. Borrelli, and K. Fregene. Decentralized receding horizon control and coordination of autonomous vehicle formations. *IEEE Transactions on Control System Technology*, 16:19–33, 2008.

[70] U. A. Khan and J. M. F. Moura. Distributing the Kalman filter for large-scale systems. *IEEE Transactions on Signal Processing*, 56:4919 – 4935, 2008.

[71] T. Knudsen, K. Trangbaek, and C.S. Kallesøe. Plug and play process control applied to a district heating system. In *Proceedings of the 17th IFAC World Congress*, Seoul, Korea, July 6-11 2008.

[72] P. Kokotovic, H. K. Khalil, and J. O'Reilly. *Singular Perturbation Methods in Control: Analysis and Design*. Academic Press, London, 1986.

[73] C. Kravaris and Y. Arkun. Geometric nonlinear control - an overview. In *Proceedings of 4th International Conference on Chemical Process Control*, pages 477–515, Y. Arkun and W. H. Ray (Eds.), Padre Island, TX, 1991.

[74] A. Kumar and P. Daoutidis. Nonlinear dynamics and control of process systems with recycle. *Journal of Process Control*, 12:475–484, 2002.

[75] J. H. Lee. Recent advances in model predictive control and other related areas. In *Proceedings of 5th International Conference on Chemical Process Control*, pages 201–216, Tahoe City, CA, 1997.

[76] S. Li, Y. Zhang, and Q. Zhu. Nash-optimization enhanced distributed model predictive control applied to the shell benchmark problem. *Information Science*, 170:329–349, 2005.

[77] D. Limon, I. Alvarado, T. Alamo, and E.F. Camacho. Robust tube-based MPC for tracking of constrained linear systems with additive disturbances. *Journal of Process Control*, 20:248 – 260, 2010.

[78] Y. Lin, E. D. Sontag, and Y. Wang. A smooth converse Lyapunov theorem for robust stability. *SIAM Journal on Control and Optimization*, 34:124–160, 1996.

[79] G.-P. Liu, Y. Xia, J. Chen, D. Rees, and W. Hu. Networked predictive control of systems with random networked delays in both forward and feedback channels. *IEEE Transactions on Industrial Electronics*, 54:1282–1297, 2007.

[80] J. Liu, X. Chen, D. Muñoz de la Peña, and P. D. Christofides. Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. *AIChE Journal*, 56:2137–2149, 2010.

[81] J. Liu, X. Chen, D. Muñoz de la Peña, and P. D. Christofides. Iterative distributed model predictive control of nonlinear systems: Handling asynchronous, delayed measurements. *IEEE Transactions on Automatic Control*, in press.

[82] J. Liu, D. Muñoz de la Peña, and P. D. Christofides. Distributed model predictive control of nonlinear process systems. *AIChE Journal*, 55:1171–1184, 2009.

[83] J. Liu, D. Muñoz de la Peña, and P. D. Christofides. Distributed model predictive control of nonlinear systems subject to asynchronous and delayed measurements. *Automatica*, 46:52–61, 2010.

[84] J. Liu, D. Muñoz de la Peña, P. D. Christofides, and J. F. Davis. Lyapunov-based model predictive control of nonlinear systems subject to time-varying measurement delays. *International Journal of Adaptive Control and Signal Processing*, 23:788–807, 2009.

[85] J. Maciejowski. *Predictive Control with Constraints*. Prentice-Hall, 2001.

[86] U. Maeder, R. Cagienard, and M. Morari. Explicit model predictive control. In S. Tarbouriech, G. Garcia, and A. H. Glattfelder, editors, *Advanced Strategies in Control Systems with Input and Output Constraints, Lecture Notes in Control and Information Sciences*, volume 346, pages 237–271. Springer-Verlag, Berlin, 2007.

[87] J. M. Maestre, D. Muñoz de la Peña, and E. F. Camacho. Distributed model predictive control based on a cooperative game. *Optimal Control Applications and Methods*, 32:153–176, 2011.

[88] J.M. Maestre, D. Muñoz de la Peña, E.F. Camacho, and T. Alamo. Distributed model predictive control based on agent negotiation. *Journal of Process Control*, 21:685–697, 2011.

[89] L. Magni, G. De Nicolao, L. Magnani, and R. Scattolini. A stabilizing model-based predictive control for nonlinear systems. *Automatica*, 37:1351–1362, 2001.

[90] L. Magni and R. Scattolini. Model predictive control of continuous-time nonlinear systems with piecewise constant control. *IEEE Transactions on Automatic Control*, 49:900–906, 2004.

[91] L. Magni and R. Scattolini. Stabilizing decentralized model predictive control of nonlinear systems. *Automatica*, 42:1231–1236, 2006.

[92] M. Mahmood and P. Mhaskar. Enhanced stability regions for model predictive control of nonlinear process systems. *AIChE J.*, 54:1487–1498, 2008.

[93] M. W. Maier. Architecting principles for systems-of-systems. *Systems Engineering*, 4:267–284, 1998.

[94] J. L. Massera. Contributions to stability theory. *Annals of Mathematics*, 64:182–206, 1956.

[95] D. Q. Mayne. Nonlinear model predictive control: An assessment. In *Proceedings of 5th International Conference on Chemical Process Control*, pages 217–231, Tahoe City, CA, 1997.

[96] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[97] D.Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 35:814–824, 1990.

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

[99] M. McKeon-Slattery. The world of wireless. *Chemical Engineering Progress*, 106:6–11, 2010.

[100] M. Mercangoz and F. J. Doyle. Distributed model predictive control of an experimental four-tank system. *Journal of Process Control*, 17:297–308, 2007.

[101] M.D. Mesarovic, D. Macko, and Y. Takahara. *Theory of Hierarchical, Multilevel, Systems*. Academic Press, New York, 1970.

[102] P. Mhaskar, N. H. El-Farra, and P. D. Christofides. Predictive control of switched nonlinear systems with scheduled mode transitions. *IEEE Transactions on Automatic Control*, 50:1670–1680, 2005.

[103] P. Mhaskar, N. H. El-Farra, and P. D. Christofides. Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control. *Systems and Control Letters*, 55:650–659, 2006.

[104] P. Mhaskar, A. Gani, C. McFall, P. D. Christofides, and J. F. Davis. Fault-tolerant control of nonlinear process systems subject to sensor faults. *AIChE Journal*, 53:654–668, 2007.

[105] A.G. Michelsen, R.I. Zamanabadi, and J. Stoustrup. Towards automatic model based controller design for reconfigurable plants. In *Proceedings of the 17th IFAC World Congress*, Seoul, Korea, 2008.

[106] N. Motee and B. Sayyar-Rodsari. Optimal partitioning in distributed model predictive control. In *Proceedings of the American Control Conference*, pages 5300–5305, Minneapolis, Minnesota, 2006.

[107] D. Muñoz de la Peña and P. D. Christofides. Lyapunov-based model predictive control of nonlinear systems subject to data losses. *IEEE Transactions on Automatic Control*, 53:2076–2089, 2008.

[108] A. G. O. Mutambara. *Decentralized estimation and control for multisensor systems*. CRC Press, 1998.

[109] P. Mhaskar N. H. El-Farra and P. D. Christofides. Uniting bounded control and MPC for stabilization of constrained linear systems. *Automatica*, 40:101–110, 2004.

[110] D. Nešić, A Teel, and P. Kokotovic. Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete time approximations. *Systems and Control Letters*, 38:259–270, 1999.

[111] D. Nešić and A. R. Teel. Input-to-state stability of networked control systems. *Automatica*, 40:2121–2128, 2004.

[112] I. Necoara, V. Nedelcu, and I. Dumitrache. Parallel and distributed optimization methods for estimation and control in networks. *Journal of Process Control*, 21:756–766, 2011.

[113] R. R. Negenborn. *Multi-Agent Model Predictive Control with Applications to Power Networks*. PhD thesis, Delft University of Technology, 2007.

[114] R. R. Negenborn, B. De Schutter, and J. Hellendoorn. Multi-agent model predictive control for transportation networks: Serial versus parallel schemes. *Engineering Applications of Artificial Intelligence*, 21:353–366, 2008.

[115] R. R. Negenborn, S. Leirens, B. De Schutter, and J. Hellendoorn. Supervisory nonlinear MPC for emergency voltage control using pattern search. *Control Engineering Practice*, 7:841–848, 2009.

[116] R. R. Negenborn, P. J. Van Overloop, T. Keviczky, and B. De Schutter. Distributed model predictive control of irrigation canals. *Networks and Heterogeneus Media*, 4:359–380, 2009.

[117] P. Neumann. Communication in industrial automation: What is going on? *Control Engineering Practice*, 15:1332–1347, 2007.

[118] C. Ocampo-Martinez, S. Bovo, and V. Puig. Partitioning approach oriented to the decentralised predictive control of large-scale systems. *Journal of Process Control*, 21:775–786, 2011.

[119] B. A. Ogunnaike and W. H. Ray. *Process Dynamics, Modeling, and Control*. Oxford University Press, New York, 1994.

[120] R. Olfati-Saber. Distributed Kalman filtering for sensor networks. *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 5492 – 5498, 2007.

[121] S. Perk, F. Teymour, and A. Cinar. Statistical monitoring of complex chemical processes using agent-based systems. *Industrial & Engineering Chemistry Research*, 49:5080–5093, 2010.

[122] B. Picasso, D. De Vito, R. Scattolini, and P. Colaneri. An MPC approach to the design of two-layer hierarchical control systems. *Automatica*, 46:823–831, 2010.

[123] G. Pin and T. Parisini. Stabilization of networked control systems by nonlinear model predictive control: A set invariance approach. In Lalo Magni, Davide Raimondo, and Frank Allgwer, editors, *Nonlinear Model Predictive Control*, volume 384 of *Lecture Notes in Control and Information Sciences*, pages 195–204. Springer Berlin / Heidelberg, 2009.

[124] D. M. Raimondo, L. Magni, and R. Scattolini. Decentralized MPC of nonlinear system: An input-to-state stability approach. *International Journal of Robust and Nonlinear Control*, 17:1651–1667, 2007.

[125] A. Rantzer. Dynamic dual decomposition for distributed control. In *Procceedings of the American Control Conference*, pages 884–888, Saint Louis, Missouri, 2009.

[126] B. S. Rao and H. F. Durrant-Whyte. Fully decentralised algorithm for multisensor Kalman filtering. *IEE Proceedings on Control Theory and Applications, D*, 138:413–420, 1991.

[127] J. B. Rawlings, D. Bonné, J. B. Jørgensen, A. N. Venkat, and S. Bay Jørgensen. Unreachable setpoints in model predictive control. *IEEE Transactions on Automatic Control*, 53:2209–2215, 2008.

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

[129] J. B. Rawlings and B. T. Stewart. Coordinating multiple optimization-based controllers: New opportunities and challenges. *Journal of Process Control*, 18:839–845, 2008.

[130] A. Richards and J. P. How. Robust distributed model predictive control. *International Journal of Control*, 80:1517–1531, 2007.

[131] J. A. Romagnoli and A. Palazoglu. *Introduction to Process Control*. CRC Press-Taylor & Francis, Boca Raton, FL, 2006.

[132] N. R. Sandell, P. Varajya, M. Athans, and M. C. Safonov. Survey of decentralized control methods for large scale systems. *IEEE Transactions on Automatic Control*, 23:108–128, 1978.

[133] R. Scattolini. A multirate self-tuning predictive controller for multi-variable systems. *International Journal of Systems Science*, 23:1347–1359, 1992.

[134] R. Scattolini. Architectures for distributed and hierarchical model predictive control - A review. *Journal of Process Control*, 19:723–731, 2009.

[135] R. Scattolini and N. Schiavoni. A multirate model based predictive controller. *IEEE Transactions on Automatic Control*, 40:1093–1097, 1995.

[136] H. Scheu, J. Busch, and W. Marquardt. Nonlinear distributed dynamic optimization based on first order sensitivities. In *Proceedings of the American Control Conference*, pages 1574–1579, 2010.

[137] H. Scheu and W. Marquardt. Sensitivity-based coordination in distributed model predictive control. *Journal of Process Control*, 21:715–728, 2011.

[138] D. E. Seborg, D. A. Mellichamp, T. F. Edgar, and F. J. Doyle. *Process Dynamics and Control*. John Wiley & Sons, New York, 2010.

[139] D. D. Siljak. *Decentralized Control of Complex Systems*. Academic Press, London, 1991.

[140] D. D. Siljak and A. I. Zecević. Control of large-scale systems: Beyond decentralized feedback. *Annual Reviews in Control*, 29:169–179, 2005.

[141] S.S. Stanković, M.S. Stanković, and D.M. Stipanović. Consensus based overlapping decentralized estimation with missing observations and communication faults. *Automatica*, 45:1397–1406, 2009.

[142] S.S. Stanković, M.S. Stanković, and D.M. Stipanović. Consensus based overlapping decentralized estimator. *IEEE Transactions on Automatic Control*, 54:410–415, 2009.

[143] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia. Coorperative distributed model predictive control. *Systems and Control Letters*, 59:460–469, 2010.

[144] B. T. Stewart, S. J. Wright, and J. B. Rawlings. Cooperative distributed model predictive control for nonlinear systems. *Journal of Process Control*, 21:698–704, 2011.

[145] J. Stoustrup. Plug and play control: control technology towards new challenges. *European Journal of Control*, 3:311–330, 2009.

[146] Y. Sun and N. H. El-Farra. Quasi-decentralized model-based networked control of process systems. *Computers and Chemical Engineering*, 32:2016–2029, 2008.

[147] A. N. Tarău, B. De Schutter, and J. Hellendoorn. Model-based control for route choice in automated baggage handling systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 40:341–351, 2010.

[148] E. Tatara, A. Cinar, and F. Teymour. Control of complex distributed systems with distributed intelligent agents. *Journal or Process Control*, 17:415–427, 2007.

[149] P. Tatjewski. Advanced control and on-line process optimization in multilayer structures. *Annual Reviews in Control*, 32:71–85, 2008.

[150] P. Trodden and A. Richards. Robust distributed model predictive control using tubes. In *Proceedings of the American Control Conference*, pages 2034–2039, Minneapolis, Minnesota, 2006.

[151] R. Vadigepalli and F.J. Doyle. A distributed state estimation and control algorithm for plantwide processes. *IEEE Transactions on Control Systems Technology*, 11:119 – 127, 2003.

[152] P. Valckenaers, H. Van Brussel, J. Wyns, P. Peeters, and L. Bongaerts. Multi-agent manufacturing control in holonic manufacturing systems. *Human Systems Management*, 18:233–243, 1999.

[153] E. J. Van Henten and J. Bontsema. Time-scale decomposition of an optimal control problem in greenhouse climate management. *Control Engineering Practice*, 17:88–96, 2009.

[154] A. N. Venkat, J. B. Rawlings, and S. J. Wright. Stability and optimality of distributed model predictive control. In *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference*, pages 6680–6685, Seville, Spain, 2005.

[155] D. De Vito, B. Picasso, and R. Scattolini. On the design of reconfigurable two-layer hierarchical control systems with MPC. In *Proceedings of the American Control Conference*, pages 4707–4712, Baltimore, Maryland, 2010.

[156] G. Walsh, O. Beldiman, and L. Bushnell. Asymptotic behavior of nonlinear networked control systems. *IEEE Transactions on Automatic Control*, 46:1093–1097, 2001.

[157] G. Walsh, H. Ye, and L. Bushnell. Stability analysis of networked control systems. *IEEE Transactions on Control Systems Technology*, 10:438–446, 2002.

[158] S.-H. Wang and E. J. Davision. On the stabilization of decentralized control systems. *IEEE Transactions on Automatic Control*, AC-18:473–478, 1973.

[159] E. B. Ydstie. Certainty equivalence adaptive control: Paradigms puzzles and switching. In *Proceedings of 5th International Conference on Chemical Process Control*, pages 9–23, Tahoe City, CA, 1997.

[160] E. B. Ydstie. New vistas for process control: Integrating physics and communication networks. *AIChE Journal*, 48:422–426, 2002.

[161] A. Zafra-Cabeza, J.M. Maestre, M.A. Ridao, E.F. Camacho, and L. Sanchez. A hierarchical distributed model predictive control approach to irrigation canals: A risk mitigation perspective. *Journal of Process Control*, 21:787–799, 2011.

[162] Y. Zhang and S. Li. Networked model predictive control based on neighbourhood optimization for serially connected large-scale processes. *Journal of Process Control*, 17:37–50, 2007.

[163] P. Zornio and B. Karschnia. Realizing the promise of wireless. *Chemical Engineering Progress*, 105:22–29, 2009.