Distributed Model Predictive Control of Nonlinear Process Systems

Jinfeng Liu
Dept. of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095

David Muñoz de la Peña
Dept. de Ingeniería de Sistemas y Automática, Universidad de Sevilla, Camino de los Descubrimientos S/N, 41092, Sevilla, Spain

Panagiotis D. Christofides
Dept. of Chemical and Biomolecular Engineering, and Dept. of Electrical Engineering, University of California, Los Angeles, CA 90095

DOI 10.1002/aic.11801
Published online April 7, 2009 in Wiley InterScience (www.interscience.wiley.com).

This work focuses on a class of nonlinear control problems that arise when new control systems which may use networked sensors and/or actuators are added to already operating control loops to improve closed-loop performance. In this case, it is desirable to design the pre-existing control system and the new control system in a way such that they coordinate their actions. To address this control problem, a distributed model predictive control method is introduced where both the pre-existing control system and the new control system are designed via Lyapunov-based model predictive control. Working with general nonlinear models of chemical processes and assuming that there exists a Lyapunov-based controller that stabilizes the nominal closed-loop system using only the pre-existing control loops, two separate Lyapunov-based model predictive controllers are designed that coordinate their actions in an efficient fashion. Specifically, the proposed distributed model predictive control design preserves the stability properties of the Lyapunov-based controller, improves the closed-loop performance, and allows handling input constraints. In addition, the proposed distributed control design requires reduced communication between the two distributed controllers since it requires that these controllers communicate only once at each sampling time and is computationally more efficient compared to the corresponding centralized model predictive control design. The theoretical results are illustrated using a chemical process example. *AIChE J,* 55: 1171–1184, 2009
Keywords: distributed model predictive control, nonlinear systems, networked control systems, process control

Introduction

The chemical process industries constitute a key economic sector in the U.S. and globally. While the range of valuable assets in a plant is large, nearly all the economic value in terms of operating profit is a direct result of plant operations. Therefore, optimal process operation and management of abnormal situations during plant operation are major challenges in the process industries since, for example, abnormal situations account for at least \$10 billion in annual lost revenue in the U.S. alone.[1] This realization has motivated extensive research in the area of chemical process control to

Correspondence concerning this article should be addressed to P. D. Christofides at pdc@seas.ucla.edu

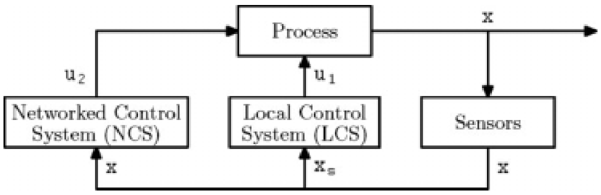

Figure 1. Decentralized networked control architecture.

ensure safe and efficient process operation. From a control architecture standpoint, control systems traditionally utilize dedicated, point-to-point wired communication links to measurement sensors and control actuators to regulate process variables at desired values. While this paradigm to process control has been successful, we are currently witnessing an augmentation of the existing, dedicated local control networks, with additional networked (wired and/or wireless) actuator/sensor devices which have become cheap and easy-to-install the last few years. Such an augmentation in sensor information and networked-based availability of data has the potential[2–6] to be transformative in the sense of dramatically improving the ability of the control systems to optimize process performance (i.e., achieving control objectives that go well beyond the ones that can be achieved with dedicated, local control systems) and prevent or deal with abnormal situations more quickly and effectively (fault-tolerance). The addition of networked sensors and actuators allows for easy modification of the control strategy by rerouting signals, having redundant systems that can be activated automatically when failures occur, and in general, they allow having improved control over the entire plant.

However, augmenting dedicated, local control systems (LCS) with control systems that may utilize real-time sensor and actuator networks gives rise to the need to design/redesign and coordinate separate control systems that operate on a process. Model predictive control (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 also 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. Motivated by the lack of available methods for the design of networked control systems (NCS) for chemical processes, in a recent work[7], we introduced a networked control architecture for nonlinear processes, shown in Figure 1. In this architecture, the local, pre-existing control system uses continuous sensing and actuation and an explicit control law (for example, the local controller is a classical controller, like a proportional-integral-derivative controller, or a nonlinear controller designed via geometric or Lyapunov-based control methods for which an explicit formula for the calculation of the control action is available). On the other hand, the NCS uses networked (wired or wireless) sensors and actuators and has access to heterogeneous, asynchronous measurements that are not available to the LCS. The NCS is designed via Lyapunov-based model predictive control (LMPC). An important feature of the NCS architecture of Figure 1 is that there is no communication between the LCS and NCS since the networked LMPC can estimate the control actions of the local controller using the explicit formula of this controller, and thus, it can take into account the actions of the local controller in the computation of its optimal input trajectories. In this sense, the networked control architecture of Figure 1 can be thought of as a decentralized one. This lack of communication is an appealing feature because the addition of the NCS does not lead to any modification of the pre-existing LCS and improves the overall robustness of the combined NCS/LCS architecture (i.e., the achievable closed-loop performance is invariant to disruptions in the communication between the NCS and LCS). Within process control, other important recent work on the subject of networked process control includes the development of a quasi-decentralized control framework for multiunit plants that achieves the desired closed-loop objectives with minimal cross communication between the plant units.[8]

Despite this progress, there are important controller design problems that remain unresolved. For example, when the LCS is a model predictive control system for which there is no explicit controller formula to calculate its future control actions, it is necessary to redesign both the NCS and the LCS and establish some, preferably small, communication between them so that they coordinate their actions. To this end, we will adopt in this work a distributed MPC approach to the design of the NCS and LCS, as shown in Figure 2. It is important to remark, at this point, that an alternative approach to address the integrated design of the NCS and LCS would be to design a fully centralized MPC to decide the manipulated inputs of all the control actuators (i.e., both u_1 and u_2 in Figure 2). However, the computational complexity of MPC grows significantly with the increase of optimization (decision) variables, which may prohibit certain online centralized MPC applications with a large number of decision variables.

With respect to available results on distributed MPC design, several distributed MPC methods have been proposed in the literature that deal with the coordination of separate MPC controllers that communicate to obtain optimal input trajectories in a distributed manner; see Refs. 9 and 10 for reviews of results in this area. In Ref. [11], the problem of distributed control of dynamically coupled nonlinear systems that are subject to decoupled constraints was considered. In Refs. 12 and 13, the effect of the coupling was modeled as a bounded disturbance compensated using a robust MPC formulation. In Ref. 14, 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. In Ref. 15, distributed MPC of decoupled systems (a class of systems of relevance in the framework of multiagents systems) was studied. In

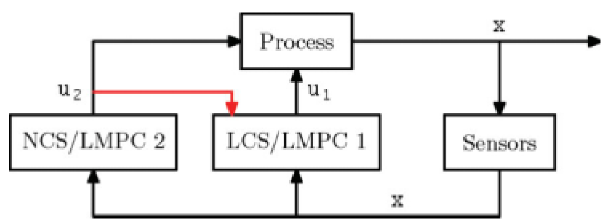

Figure 2. Distributed LMPC control architecture for networked control system design.

[Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Ref. 16, an MPC algorithm was proposed under the main assumptions that the system is nonlinear, discrete-time and no information is exchanged between local controllers, and in Ref. 17, MPC for nonlinear systems was studied from an input-to-state stability point of view.

In the present work, we introduce a distributed model predictive control method for the design of networked control systems where both the pre-existing local control system and the networked control system are designed via Lyapunov-based model predictive control. The proposed distributed MPC design—see Figure 2—uses a hierarchical control architecture in the sense that the LCS stabilizes the closed-loop system and the NCS takes advantage of additional control inputs to improve the closed-loop performance. This hierarchically distributed MPC design is different from previous distributed MPC designs which decompose a centralized control problem spatially. In particular, the proposed design provides the potential of maintaining stability and performance in the face of new/failing actuators, (for example, the failure of the actuator of the NCS (zero input) does not affect the closed-loop stability). Working with general nonlinear models of chemical processes and assuming that there exists a Lyapunov-based controller that stabilizes the nominal closed-loop system using only the pre-existing control loops, two separate Lyapunov-based model predictive controllers are designed that coordinate their actions in an efficient fashion. Specifically, the proposed distributed MPC design preserves the stability properties of the Lyapunov-based controller, improves the closed-loop performance and allows handling input constraints. In addition, the proposed distributed control design requires reduced communication between the two distributed controllers since it requires that these controllers communicate only once at each sampling time and is computationally more efficient compared to the corresponding centralized MPC design. The theoretical results are illustrated using a chemical process example.

Preliminaries

Problem formulation

We consider nonlinear process systems described by the following state-space model

$$\dot{x}(t) = f(x(t), u_1(t), u_2(t), w(t)) \qquad (1)$$

where $x(t) \in R^{n_x}$ denotes the vector of process state variables, $u_1(t) \in R^{n_{u1}}$ and $u_2(t) \in R^{n_{u2}}$ are two separate sets of control (manipulated) inputs and $w(t) \in R^{n_w}$ denotes the vector of disturbance variables. The two inputs are restricted to be in two nonempty convex sets $U_1 \subseteq R^{n_{u1}}$ and $U_2 \subseteq R^{n_{u2}}$ and the disturbance vector is bounded, i.e., $w(t) \in W$ where

$$W := \{w \in R^{n_w} \text{s.t.} |w| \leq \theta, \theta > 0\}.^*$$

We assume that f is a locally Lipschitz vector function and $f(0,0,0,0) = 0$. This means that the origin is an equilibrium point for the nominal system (system of Eq. 1 with $w(t) = 0$ for all t) with $u_1 = 0$ and $u_2 = 0$. System of Eq. 1 is controlled with the two sets of manipulated inputs u_1 and u_2, which could be multiple inputs of a system or a single input divided artificially into two terms (i.e., $\dot{x}(t) = \hat{f}(x(t), u(t), w(t))$ with $u(t) = u_1(t) + u_2(t)$). We also assume that the state x of the system is sampled synchronously and continuously and the time instants in which we have measurement samplings are 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.

Remark 1. In general, distributed control systems are formulated based on the assumption that the controlled systems are decoupled or partially decoupled. However, we consider a fully coupled process model with two sets of possible manipulated inputs; this is a very common occurrence in chemical process control as we will illustrate in the example of "Application to a Reactor-Separator Process" section. It is important to note that even though we have motivated the control problem of Eq. 1 by the augmentation of LCS with NCS, the same control formulation could be used when a new control system which may use a local control network is added to a process that already operates under an MPC; see example in "Application to a Reactor-Separator Process" section.

Remark 2. We have considered that the full state measurements are available to simplify the notation. The results can be extended to controllers based on partial state measurement, continuous/asynchronous measurements, continuous/delayed measurements, but the complete theoretical development for these cases is outside the scope of the present manuscript.

Lyapunov-based controller

We assume that there exists a Lyapunov-based controller $u_1(t) = h(x(t))$ which satisfies the input constraint on u_1 for all x inside a given stability region and renders the origin of the nominal closed-loop system asymptotically stable with $u_2(t) = 0$. Using converse Lyapunov theorems, this assumption implies that there exist functions $\alpha_i(\cdot)$, $i = 1,2,3,4$ of class $\mathcal{K}^\dagger$ and a control Lyapunov function V for the nominal closed-loop system which is continuous and bounded in R^{n_x}, that satisfy the following inequalities

$$\alpha_1(|x|) \leq V(x) \leq \alpha_2(|x|)$$
$$\frac{\partial V(x)}{\partial x} f(x, h(x), 0, 0) \leq -\alpha_3(|x|)$$
$$\left|\frac{\partial V(x)}{\partial x}\right| \leq \alpha_4(|x|)$$
$$h(x) \in U_1$$

for all $x \in D \subseteq R^{n_x}$ where D is an open neighborhood of the origin. We denote the region $\Omega_\rho{}^\ddagger \subseteq D$ as the stability region of the closed-loop system under the control $u_1 = h(x)$ and $u_2 = 0$.

By continuity, the local Lipschitz property assumed for the vector field $f(x,u_1,u_2,w)$ and the fact that the manipulated inputs u_1 and u_2 are bounded in convex sets, there exists a positive constant M such that

$$|f(x, u_1, u_2, w)| \leq M \qquad (3)$$

for all $x \in \Omega_\rho$, $u_1 \in U_1$, $u_2 \in U_2$ and $w \in W$. In addition, by the continuous differentiable property of the Lyapunov function V

*$|\cdot|$ denotes Euclidean norm of a vector.

†A continuous function $\alpha : [0,a) \to [0,\infty)$ is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$.

‡We use Ω_r to denote the set $\Omega_r := \{x \in R^{n_x} | V(x) \leq r\}$.

and the Lipschitz property assumed for the vector field $f(x,u_1,u_2,w)$, there exist positive constants L_x, L_w such that

$$\left|\frac{\partial V}{\partial x}f(x,u_1,u_2,w)-\frac{\partial V}{\partial x}f(x',u_1,u_2,0)\right|\le L_x|x-x'|+L_w|w|. \tag{4}$$

for all $x,x' \in \Omega_\rho$, $u_1 \in U_1$, $u_2 \in U_2$, and $w \in W$. These constants will be used in the proof of Theorem 1 in "Distributed LMPC stability" section.

Remark 3. The assumption that there exists a controller $u_1 = h(x)$ which can stabilize the closed-loop system with $u_2 = 0$ implies that, in principle, it is not necessary to use the extra input u_2 in order to achieve closed-loop stability. However, one of the main objectives of the proposed distributed control method is to profit from the extra control effort to improve the closed-loop performance while maintaining the stability properties achieved by only implementing u_1.

Remark 4. Different state feedback control laws for nonlinear systems have been developed using Lyapunov techniques; the reader may refer to Refs. 18–20 for results in this area including results on the design of bounded Lyapunov-based controllers by taking explicitly into account input constraints for broad classes of nonlinear systems. In the example of "Application to a Reactor-Separator Process" section, we will use the Lyapunov-based feedback control law proposed in Ref. 21 (see also Refs. 22 and 23) which is based on a control Lyapunov function of the open-loop system.

Remark 5. Note that while there are currently no general methods for constructing Lyapunov functions for general nonlinear systems, for broad classes of nonlinear models arising in the context of chemical process control applications, quadratic Lyapunov functions are widely used and provide very good estimates of closed-loop stability regions.

Centralized Lyapunov-based MPC

To take advantage of both sets of manipulated inputs u_1 and u_2, one option is to design a centralized MPC. To guarantee robust stability of the closed-loop system, the MPC must include a set of stability constraints. To do this, we propose to use the LMPC method proposed in Refs. 24 and 25 which guarantees practical stability of the closed-loop system, allows for an explicit characterization of the stability region and gives a reduced complexity optimization problem. LMPC is based on uniting receding horizon control with Lyapunov functions and computes the manipulated input trajectory solving a finite horizon constrained optimal control problem. The LMPC controller is based on the previously designed Lyapunov-based controller h. The controller h is used to define a contractive constraint for the LMPC method which guarantees that the LMPC inherits the stability and robustness properties of the Lyapunov-based controller. The LMPC method introduced in Refs. 24 and 25 is based on the following optimization problem

$$\min_{u_{c1},u_{c2}\in S(\Delta)} \int_0^{N\Delta} \left[\tilde{x}^T(\tau)Q_c\tilde{x}(\tau)+u_{c1}^T(\tau)R_{c1}u_{c1}(\tau)+u_{c2}^T(\tau)R_{c2}u_{c2}(\tau)\right]d\tau \tag{5a}$$

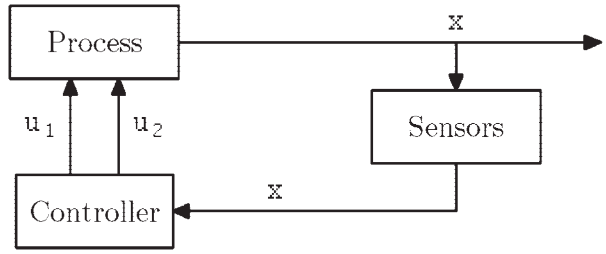

Figure 3. Centralized control system.

$$\text{s.t. } \dot{\tilde{x}}(\tau)=f(\tilde{x}(\tau),u_{c1}(\tau),u_{c2}(\tau),0) \tag{5b}$$

$$\tilde{x}(0)=x(t_k) \tag{5c}$$

$$u_{c1}(\tau)\in U_1 \tag{5d}$$

$$u_{c2}(\tau)\in U_2 \tag{5e}$$

$$\frac{\partial V(x)}{\partial x}f(x(t_k),u_{c1}(0),u_{c2}(0),0)\le\frac{\partial V(x)}{\partial x}f(x(t_k),h(x(t_k)),0,0) \tag{5f}$$

where $S(\Delta)$ is the family of piece-wise constant functions with sampling period Δ, Q_c, R_{c1}, and R_{c2} are positive definite weight matrices that define the cost, $x(t_k)$ is the state measurement obtained at t_k, $\tilde{x}$ is the predicted trajectory of the nominal system for the input trajectory computed by the LMPC, N is the prediction horizon and V is the Lyapunov function corresponding to the controller $h(x)$.

The optimal solution to this optimization problem is denoted by $u_{c1}^*(\tau|t_k)$ and $u_{c2}^*(\tau|t_k)$. The LMPC controller is implemented with a receding horizon method; that is, at each sampling time t_k, the new state $x(t_k)$ is received from the sensors, the optimization problem of Eq. 5 is solved, and $u_{c1}^*(t-t_k|t_k)$ and $u_{c2}^*(t-t_k|t_k)$ are applied to the closed-loop system for $t \in [t_k, t_{k+1})$. In what follows, we refer to this controller as the centralized LMPC. Figure 3 shows a schematic of this kind of control system.

The optimization problem of Eq. 5 does not depend on the uncertainty and assures that the system in closed-loop with the LMPC controller of Eq. 5 maintains the stability properties of the Lyapunov-based controller $u_1 = h(x)$ with $u_2 = 0$. The contractive constraint of Eq. 5f guarantees that the value of the time derivative of the Lyapunov function at the initial evaluation time of the centralized LMPC is lower or equal to the value obtained if only the Lyapunov-based controller $u_1 = h(x(t_k))$ is implemented in the closed-loop system. This is the contractive constraint that allows one to prove that the centralized LMPC inherits the stability and robustness properties of the Lyapunov-based controller h.

The manipulated inputs of the closed-loop system under the above centralized LMPC are defined as follows

$$\begin{aligned} u_1(t)&=u_{c1}^*(t-t_k|t_k),\forall t\in[t_k,t_{k+1})\\ u_2(t)&=u_{c2}^*(t-t_k|t_k),\forall t\in[t_k,t_{k+1}). \end{aligned} \tag{6}$$

The main property of the centralized LMPC is that the origin of the closed-loop system is practically stable for all initial states inside the stability region Ω_ρ for a sufficient small sampling time Δ and disturbance upper bound θ. This property is also guaranteed by the Lyapunov-based controller when this controller is implemented in a sample-and-hold fashion (see Refs. 26 and 27 for results on sampled-data systems). The main advantage of LMPC approaches with respect to the Lyapunov-based controller is that optimality considerations can be taken explicitly into account (as well as constraints on the inputs and the states[25]) in the computation of the controller within an online optimization framework while improving the closed-loop performance.

Distributed LMPC

Distributed LMPC formulations

The objectives of this work are to propose a control architecture that has the potential to maintain the closed-loop stability and performance in face of new or failing actuators and to reduce the computational burden in the evaluation of the optimal manipulated inputs for systems with a high-number of control inputs that can be divided into two sets. Note that in general, the coordination of two controllers to regulate the same process is a difficult problem. In the present work, we design two separate LMPCs to compute u_1 and u_2 and refer to the LMPC computing the trajectories of u_1 and u_2 as LMPC 1 and LMPC 2, respectively. Figure 2 shows a schematic of the proposed distributed method. We propose to use the following implementation strategy:

(1) At each sampling instant t_k, both LMPC 1 and LMPC 2 receive the state measurement $x(t_k)$ from the sensors.

(2) LMPC 2 evaluates the optimal input trajectory of u_2 based on the current state measurement and sends the first step input value to its corresponding actuators and the entire optimal input trajectory to LMPC 1.

(3) Once LMPC 1 receives the entire optimal input trajectory for u_2 from LMPC 2, it evaluates the future input trajectory of u_1 based on $x(t_k)$ and the entire optimal input trajectory of u_2 computed by LMPC 2.

(4) LMPC 1 sends the first step input value of u_1 to the corresponding actuators.

First we define the optimization problem of LMPC 2. This optimization problem depends on the latest state measurement $x(t_k)$, however, LMPC 2 does not have any information about the value that u_1 will take. In order to make a decision, LMPC 2 must assume a trajectory for u_1 along the prediction horizon. To this end, the Lyapunov-based controller $u_1 = h(x)$ is used. To inherit the stability properties of this controller, u_2 must satisfy a contractive constraint that guarantees a given minimum decrease rate of the Lyapunov function V. The proposed LMPC 2 is based on the following optimization problem:

$$\min_{u_{d2}\in S(\Delta)} \int_0^{N\Delta} \left[\tilde{x}^T(\tau)Q_c\tilde{x}(\tau) + u_{d1}^T(\tau)R_{c1}u_{d1} + u_{d2}^T(\tau)R_{c2}u_{d2}(\tau)\right]d\tau \tag{7a}$$

$$\dot{\tilde{x}}(\tau) = f(\tilde{x}(\tau), u_{d1}(\tau), u_{d2}(\tau), 0) \tag{7b}$$

$$u_{d1}(\tau) = h(\tilde{x}(j\Delta)), \forall \tau \in [j\Delta, (j+1)\Delta), j = 0, \ldots, N-1 \tag{7c}$$

$$\tilde{x}(0) = x(t_k) \tag{7d}$$

$$u_{d2}(\tau) \in U_2 \tag{7e}$$

$$\frac{\partial V(x)}{\partial x} f(x(t_k), h(x(t_k)), u_{d2}(0), 0) \leq \frac{\partial V(x)}{\partial x} f(x(t_k), h(x(t_k)), 0, 0) \tag{7f}$$

where $\tilde{x}$ is the predicted trajectory of the nominal system with u_2 being the input trajectory computed by the LMPC of Eq. 7 (i.e., LMPC 2) and u_1 being the Lyapunov-based controller $h(x(t_k))$ applied in a sample and hold fashion. The optimal solution to this optimization problem is denoted by $u_{d2}^*(\tau|t_k)$. This information is sent to LMPC 1. The constraint of Eq. 7e defines the constraint on the manipulated input u_2 and the contractive constraint of Eq. 7f guarantees that the value of the time derivative of the Lyapunov function at the initial evaluation time, if $u_1 = h(x(t_k))$ and $u_2 = u_{d2}^*(0|t_k)$ are applied, is lower or equal to the value obtained when $u_1 = h(x)$ and $u_2 = 0$ are applied.

The optimization problem of LMPC 1 depends on the latest state measurement $x(t_k)$ and the decision taken by LMPC 2 (i.e., $u_{d2}^*(\tau|t_k)$). This allows LMPC 1 to compute an input u_1 such that the closed-loop performance is optimized, while guaranteeing that the stability properties of the Lyapunov-based controller are preserved. Specifically, LMPC 1 is based on the following optimization problem:

$$\min_{u_{d1}\in S(\Delta)} \int_0^{N\Delta} \left[\tilde{x}^T(\tau)Q_c\tilde{x}(\tau) + u_{d1}^T(\tau)R_{c1}u_{d1} + u_{d2}^{*T}(\tau|t_k)R_{c2}u_{d2}^*(\tau|t_k)\right]d\tau \tag{8a}$$

$$\dot{\tilde{x}}(\tau) = f\left(\tilde{x}(\tau), u_{d1}(\tau), u_{d2}^*(\tau|t_k), 0\right) \tag{8b}$$

$$\tilde{x}(0) = x(t_k) \tag{8c}$$

$$u_{d1}(\tau) \in U_1 \tag{8d}$$

$$\frac{\partial V(x)}{\partial x} f\left(x(t_k), u_{d1}(0), u_{d2}^*(0|t_k), 0\right) \leq \frac{\partial V(x)}{\partial x} f(x(t_k), h(x(t_k)), u_{d2}^*(0|t_k), 0) \tag{8e}$$

where $\tilde{x}$ is the predicted trajectory of the nominal system with u_2 being the optimal input trajectory $u_{d2}^*(\tau|t_k)$ computed by LMPC 2 and u_1 being the input trajectory computed by the LMPC of Eq. 8d (i.e., LMPC 1). The optimal solution to this optimization problem is denoted by $u_{d1}^*(\tau|t_k)$. The constraint of Eq. 8d defines the constraint on the manipulated input u_1 and the contractive constraint of Eq. 8e guarantees that the value of the time derivative of the control Lyapunov function at the initial evaluation time, if $u_1 = u_{d1}^*(0|t_k)$ and $u_2 = u_{d2}^*(0|t_k)$ are applied, is lower or equal to the value obtained when $u_1 = h(x(t_k))$ and $u_2 = u_{d2}^*(0|t_k)$ are applied.

Once both optimization problems are solved, the manipulated inputs of the proposed distributed LMPC design based on the above LMPC 1 and LMPC 2 are defined as follows:

$$\begin{aligned} u_1(t) &= u_{d1}^*(t - t_k|t_k), \forall t \in [t_k, t_{k+1}) \\ u_2(t) &= u_{d2}^*(t - t_k|t_k), \forall t \in [t_k, t_{k+1}). \end{aligned} \tag{9}$$

Remark 6. We do not consider delays introduced in the system by the communication network or by the time needed to solve each of the LMPC optimization problems. In future works, these delays will be taken into account in the formulation of the controllers.

Remark 7. At step 2 of the proposed implementation strategy, the whole optimal input trajectory of LMPC 2 is sent to LMPC 1. From the stability point of view, it is unnecessary to send the whole optimal input trajectory. Only the first step of the optimal input trajectory of LMPC 2 is needed to send to LMPC 1 in order to guarantee the stability of the closed-loop system under the distributed LMPC method (please see "Distributed LMPC stability" section for the proof of the closed-loop stability). Thus, the communication between the two LMPCs can be minimized by only sending the first step of an optimal input trajectory without loss of the closed-loop stability. However, the transmission of the whole optimal trajectory at a sampling time can, to some extend, improve the closed-loop performance because LMPC 1 has more information on the possible future input trajectory of LMPC 2.

Remark 8. The key idea of the proposed distributed LMPC formulation is to impose a hierarchy on the order in which the controllers are evaluated in order to guarantee that the resulting control action stabilizes the system. In this paper, we assume flawless communications. If data losses are taken into account, the control method has to be modified because at each time step coordination between both LMPCs is not guaranteed.

Remark 9. Since the computational burden of nonlinear MPC methods is usually high, the proposed distributed LMPC design only requires LMPC2 and LMPC 1 to "talk" once every sampling time (that is, LMPC 2 sends its optimal input trajectory to LMPC 1) to minimize the communication between the two LMPCs. This strategy is more robust when communication between the distributed LMPCs can be subject to disruption.

Remark 10. Constraints of Eqs. 7f, 8b and 8e are a key element of the distributed LMPC design. In general, guaranteeing closed-loop stability of a distributed control method is a difficult task because of the interactions between the separate controllers and can only be done under certain assumptions (see, for example, Refs. 9 and 10). Constraint of Eq. 8b guarantees that LMPC 1 takes into account the effect of LMPC 2 to the applied inputs (recall that LMPC 2 is designed without taking LMPC 1 into account). Constraints of Eqs 7f and 8e together with the hierarchical control strategy (i.e., LMPC 2 is solved first and LMPC 1 is solved second) guarantee that the value of the Lyapunov function of the closed-loop system is a decreasing sequence of time with a lower bound.

Remark 11. Note that the stability of the closed-loop system is inherited from the Lyapunov-based controller $u_1 = h(x)$. Once the contractive constraints of Eqs. 7f and 8e are satisfied, the closed-loop stability is guaranteed. The main purpose of LMPC 1 and LMPC 2 is to optimize the inputs u_1 and u_2. Thus, during the evaluation of the optimal solutions of LMPC 1 and LMPC 2 within a sampling period, we can terminate the optimization (i.e., limit the function evaluation times in the process of searching for the optimal solutions) to obtain suboptimal input trajectories without loss of the closed-loop stability. An extreme application of this idea is when the optimization process is terminated at the beginning of every optimization process which gives the inputs: $u_1(t) = h(x(t_k))$ and $u_2(t) = 0$ for $t \in [t_k, t_{k+1})$, which guarantees stability of the closed-loop system but not optimal performance.

Remark 12. In the distributed LMPC design, LMPC 2 and LMPC 1 are evaluated in sequence, which implies that the minimal sampling time of the system should be greater than or equal to the sum of the evaluation times of LMPC 2 and LMPC 1. To achieve that the two distributed LMPC optimization problems can be solved in parallel, LMPC 1 can use old input trajectories of LMPC 2, that is, at t_k, LMPC 1 uses $u_2^*(t-t_{k-1}|t_{k-1})$ to define its optimization problem. This strategy may introduce extra errors in the optimization problem, however, and may not guarantee closed-loop stability.

Remark 13. In this work, state constraints have not been considered but the proposed distributed LMPC approach can be extended to handle state constraints by restricting the closed-loop stability region further to satisfy the state constraints; please see Ref. [25] for more results on this issue.

Remark 14. The contractive constraints of Eqs. 7f and 8e guarantee that the choice of u_2 cannot render LMPC 1 infeasible. In addition, the two constraints guarantee that the proposed scheme inherits the stability region of the Lyapunov-based controller *h(x)*.

Distributed LMPC stability

Statement of Stability Results and Discussion. In this subsection, we present the stability properties of the distributed LMPC design. The proposed distributed LMPC method computes the inputs u_1 and u_2 applied to the system in a way such that in the closed-loop system, the value of the Lyapunov function at time instant t_k (i.e., $V(x(t_k))$) is a decreasing sequence of values with a lower bound. Following Lyapunov arguments, this property guarantees practical stability of the closed-loop system. This is achieved due to the contractive constraints of Eqs. 7f and 8e. This property is presented in Theorem 1 below.

Theorem 1. *Consider system of Eq. 1 in closed-loop under the distributed LMPC design of Eq. 9 based on a controller $u_1 = h(x)$ that satisfies the conditions of Eq. 2. Let $\varepsilon_w > 0$, $\Delta > 0$ and $\rho > \rho_s > 0$ satisfy the following constraint:*

$$-\alpha_3\left(\alpha_2^{-1}(\rho_s)\right) + L_x M \Delta + L_w \theta \leq -\epsilon_w/\Delta. \tag{10}$$

If $x(t_0) \in \Omega_\rho$ and if $\rho^* \leq \rho$ where

$$\rho^* = \max\{V(x(t+\Delta)) : V(x(t)) \leq \rho_s\},$$

then the state x(t) of the closed-loop system is ultimately bounded in Ω_{ρ^*}.

Proof of Theorem 1. **Proof.** The proof consists of two parts. We first prove that the optimization problems of Eqs. 7 and 8 are feasible for all states $x \in \Omega_\rho$. Then we prove that, under the distributed LMPC design of Eq. 9, the state of the system of Eq. 1 is ultimately bounded in a region that contains the origin.

Part 1: We prove the feasibility of LMPC 2 first, and then the feasibility of LMPC 1. All input trajectories of $u_2(\tau)$ such that $u_2(\tau) = 0$, $\forall \tau \in [0, \Delta)$ satisfy all the constraints (including the input constraint of Eq. 7e and contractive constraint of Eq. 7f) of LMPC 2, thus the feasibility of LMPC 2 is guaranteed. The feasibility of LMPC 1 follows because all input trajectories $u_1(\tau)$ such that $u_1(\tau) = h(x(t_k))$, $\forall \tau \in [0,\Delta)$ are feasible solutions to the optimization problem of LMPC 1 since all such trajectories satisfy the input constraint of Eq. 8d; this is guaranteed by the closed-loop stability property of the Lyapunov-based controller h and the contractive constraint of Eq. 8e.

Part 2: From conditions of Eq. 2 and the constraints of Eqs. 7f and 8e, if $x(t_k) \in \Omega_\rho$ it follows that:

$$\begin{aligned}\frac{\partial V(x)}{\partial x} f\left(x(t_k), u^*_{d1}(0|t_k), u^*_{d2}(0|t_k), 0\right) &\le \frac{\partial V(x)}{\partial x} f\left(x(t_k), h(x(t_k)), u^*_{d2}(0|t_k), 0\right) \\ &\le \frac{\partial V(x)}{\partial x} f(x(t_k), h(x(t_k)), 0, 0) \\ &\le -\alpha_3(|x(t_k)|). \end{aligned} \tag{11}$$

The time derivative of the Lyapunov function along the actual state trajectory $x(t)$ of system of Eq. 1 in $t \in [t_k, t_{k+1})$ is given by:

$$\dot{V}(x(t)) = \frac{\partial V}{\partial x} f(x(t), u^*_{d1}(0|t_k), u^*_{d2}(0|t_k), w(t)).$$

Adding and subtracting $\frac{\partial V}{\partial x} f(x(t_k), u^*_{d1}(0|t_k), u^*_{d2}(0|t_k), 0)$ and taking into account the Eq. 11, we obtain the following inequality:

$$\begin{aligned}\dot{V}(x(t)) \le -\alpha_3(|x(t_k)|) &+ \frac{\partial V(x)}{\partial x} f\left(x(t), u^*_{d1}(\tau|t_k), u^*_{d2}(\tau + t_k), w(t)\right) \\ &- \frac{\partial V}{\partial x} f\left(x(t_k), u^*_{d1}(0|t_k), u^*_{d2}(0|t_k), 0\right). \end{aligned} \tag{12}$$

From Eqs. 2, 4, and 12, the following inequality is obtained for all $x(t_k) \in \Omega_\rho/\Omega_{\rho_s}$:

$$\dot{V}(x(t)) \quad \le \quad -\alpha_3(\alpha_2^{-1}(\rho_s)) + L_x|x(t) - x(t_k)| + L_w|w|.$$

Taking into account Eq. 3 and the continuity of $x(t)$, the following bound can be written for all $t \in [t_k, t_{k+1})$

$$|x(t) - x(t_k)| \le M\Delta.$$

Using this expression, we obtain the following bound on the time derivative of the Lyapunov function for $t \in [t_k, t_{k+1})$, for all initial states $x(t_k) \in \Omega_\rho/\Omega_{\rho_s}$:

$$\dot{V}(x(t)) \le -\alpha_3(\alpha_2^{-1}(\rho_s)) + L_x M\Delta + L_w\theta.$$

If condition of Eq. 10 is satisfied, then there exists $\varepsilon_w > 0$ such that the following inequality holds for $x(t_k) \in \Omega_\rho/\Omega_{\rho_s}$:

$$\dot{V}(x(t)) \le -\epsilon_w/\Delta$$

in $t = [t_k, t_{k+1})$. Integrating this bound on $t \in [t_k, t_{k+1})$, we obtain that:

$$\begin{aligned} V(x(t_{k+1})) &\le V(x(t_k)) - \epsilon_w \\ V(x(t)) &\le V(x(t_k)), \forall t \in [t_k, t_{k+1}) \end{aligned} \tag{13}$$

for all $x(t_k) \in \Omega_\rho/\Omega_{\rho_s}$. Using Eq. 13 recursively it is proved that, if $x(t_0) \in \Omega_\rho/\Omega_{\rho_s}$, the state converges to Ω_{ρ_s} in a finite number of sampling times without leaving the stability region. Once the state converges to $\Omega_{\rho_s} \subseteq \Omega_{\rho^*}$, it remains inside Ω_{ρ^*} for all times. This statement holds because of the definition of ρ^*. This proves that the closed-loop system under the distributed LMPC design is ultimately bounded in Ω_{ρ^*}.

Remark 15. Referring to Theorem 1, condition of Eq. 10 guarantees that if the state of the closed-loop system at a sampling time t_k is outside the level set $V(x(t_k)) = \rho_s$ but inside the level set $V(x(t_k)) = \rho$, the derivative of the Lyapunov function of the state of the closed-loop system is negative under the distributed LMPC design.

Remark 18. For continuous-time systems under continuous control implementation, a sufficient condition for invariance is that the derivative of a Lyapunov function is negative on the boundary of a set. For systems with continuous-time dynamics and sample-and-hold control implementation, this condition is not sufficient because the derivative may become positive during the sampling period and the system may leave the set before a new sample is obtained. Referring to Theorem 1, ρ^* is the maximum value that the Lyapunov function can achieve in a time period of length Δ when $x(t_k) \in \Omega_{\rho_s}$. Ω_{ρ^*} defines an invariant set for the state $x(t)$ under sample-and-hold implementation of the inputs of the distributed LMPC design.

Remark 17. Although the proof of Theorem 1 is constructive, the constants obtained are conservative. In practice, the sampling time and disturbance upper bound are better estimated through closed-loop simulations. Condition of Eq. 10 is more useful as a guideline on the interaction between the various parameters that define the system and may be used as a guideline to design the distributed controllers, see Refs. 27 and 28 for further discussion on this issue.

Application to a Reactor-Separator Process

The process considered in this example is a three vessel, reactor-separator process consisting of two continuously stirred tank reactors (CSTRs) and a flash tank separator (see Figure 4). A feed stream to the first CSTR F_{10} contains the reactant A which is converted into the desired product B. The desired product B can then further react into an undesired side-product C. The effluent of the first CSTR along with additional fresh feed F_{20} makes up the inlet to the second CSTR. The reactions $A \rightarrow B$ and $B \rightarrow C$ (referred to as 1 and 2, respectively) take place in the two CSTRs in series before the effluent from CSTR 2 is fed to a flash tank. The overhead vapor from the flash tank is condensed and recycled to the first CSTR and the bottom product stream is removed. A small portion of the overhead is purged before being recycled to the first CSTR. All the three vessels are assumed to have static holdup. The dynamic equations describing the behavior of the system, obtained through

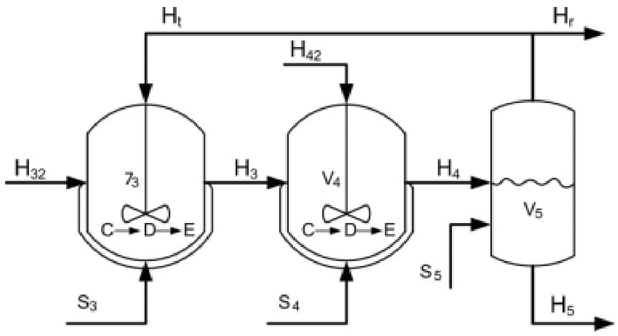

Figure 4. Reactor-separator process.

material and energy balances under standard modeling assumptions, are given below:

$$
\begin{aligned}
\frac{dx_{A1}}{dt} &= \frac{F_{10}}{V_1}(x_{A10}-x_{A1}) + \frac{F_r}{V_1}(x_{Ar}-x_{A1}) - k_1 e^{\frac{-E_1}{RT_1}} x_{A1} \\
\frac{dx_{B1}}{dt} &= \frac{F_{10}}{V_1}(x_{B10}-x_{B1}) + \frac{F_r}{V_1}(x_{Br}-x_{B1}) \\
&\quad + k_1 e^{\frac{-E_1}{RT_1}} x_{A1} - k_2 e^{\frac{-E_2}{RT_1}} x_{B1} \\
\frac{dT_1}{dt} &= \frac{F_{10}}{V_1}(T_{10}-T_1) + \frac{F_r}{V_1}(T_3-T_1) + \frac{-\Delta H_1}{C_p} k_1 e^{\frac{-E_1}{RT_1}} x_{A1} \\
&\quad + \frac{-\Delta H_2}{C_p} k_2 e^{\frac{-E_2}{RT_1}} x_{B1} + \frac{Q_1}{\rho C_p V_1} \\
\frac{dx_{A2}}{dt} &= \frac{F_1}{V_2}(x_{A1}-x_{A2}) + \frac{F_{20}}{V_2}(x_{A20}-x_{A2}) - k_1 e^{\frac{-E_1}{RT_2}} x_{A2} \\
\frac{dx_{B2}}{dt} &= \frac{F_1}{V_2}(x_{B1}-x_{B2}) + \frac{F_{20}}{V_2}(x_{B20}-x_{B2}) \\
&\quad + k_1 e^{\frac{-E_1}{RT_2}} x_{A2} - k_2 e^{\frac{-E_2}{RT_2}} x_{B2} \\
\frac{dT_2}{dt} &= \frac{F_1}{V_2}(T_1-T_2) + \frac{F_{20}}{V_2}(T_{20}-T_2) + \frac{-\Delta H_1}{C_p} k_1 e^{\frac{-E_1}{RT_2}} x_{A2} \\
&\quad + \frac{-\Delta H_2}{C_p} k_2 e^{\frac{-E_2}{RT_2}} x_{B2} + \frac{Q_2}{\rho C_p V_2} \\
\frac{dx_{A3}}{dt} &= \frac{F_2}{V_3}(x_{A2}-x_{A3}) - \frac{F_r+F_p}{V_3}(x_{Ar}-x_{A3}) \\
\frac{dx_{B3}}{dt} &= \frac{F_2}{V_3}(x_{B2}-x_{B3}) - \frac{F_r+F_p}{V_3}(x_{Br}-x_{B3}) \\
\frac{dT_3}{dt} &= \frac{F_2}{V_3}(T_2-T_3) + \frac{Q_3}{\rho C_p V_3}
\end{aligned} \tag{14}
$$

where the definitions for the variables can be found in Table 1. The model of the flash tank separator was derived under the assumption that the relative volatility for each of the species remains constant within the operating temperature range of the flash tank. This assumption allows calculating the mass fractions in the overhead based upon the mass fractions in the liquid portion of the vessel. It has also been assumed that there is a negligible amount of reaction taking place in the separator. The following algebraic equations model the composition of the overhead stream relative to the composition of the liquid holdup in the flash tank:

$$
\begin{aligned}
x_{Ar} &= \frac{\alpha_A x_{A3}}{\alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C x_{C3}} \\
x_{Br} &= \frac{\alpha_B x_{B3}}{\alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C x_{C3}} \\
x_{Cr} &= \frac{\alpha_C x_{C3}}{\alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C x_{C3}}
\end{aligned} \tag{15}
$$

Each of the tanks has an external heat input. The manipulated inputs to the system are the heat inputs to the three vessels, Q_1, Q_2, and Q_3, and the feed stream flow rate to vessel 2, F_{20}.

The process of Eq. 14 was numerically simulated using a standard Euler integration method. Process noise was added to the right-hand side of each equation in the process of Eq. 14 to simulate disturbances/model uncertainty and it was generated as autocorrelated noise of the form $w_k = \phi w_{k-1} + \xi_k$ where $k = 0,1,\ldots$ is the discrete time step of 0.001 hr, ξ_k is generated by a normally distributed random variable with standard deviation σ_p, and ϕ is the autocorrelation factor and w_k is bounded by θ_p, that is $|w_k| \leq \theta_p$. Table 2 contains the parameters used in generating the process noise.

We assume that the measurements of the temperatures T_1, T_2, T_3 and the measurements of mass fractions x_{A1}, x_{B1}, x_{A2}, x_{B2}, x_{A3}, x_{B3} are available synchronously and continuously at time instants $\{t_{k\geq 0}\}$ 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 the simulations carried out in this section, we pick the initial time to be $t_0 = 0$ and the sampling time to be $\Delta = 0.02$ h = 1.2 min.

The control objective is to regulate the system to the stable steady-state x_s corresponding to the operating point defined by Q_{1s}, Q_{2s}, Q_{3s} of u_{1s} and F_{20s} of u_{2s}. The steady-state values for u_{1s} and u_{2s} and the values of the steady-state are given in Tables 3 and 4, respectively. Taking this control objective into account, the process model of Eq. 14 belongs to the following class of nonlinear systems:

$$\dot{x}(t) = f(x(t)) + g_1(x(t))u_1(t) + g_2(x(t))u_2(t) + w(t)$$

Table 1. Process Variables

x_{A1}, x_{A2}, x_{A3}	Mass fractions of A in vessels 1, 2, 3
x_{B1}, x_{B2}, x_{B3}	Mass fractions of B in vessels 1, 2, 3
x_{C3}	Mass fraction of C in vessel 3
x_{Ar}, x_{Br}, x_{Cr}	Mass fractions of A, B, C in the recycle
T_1, T_2, T_3	Temperatures in vessels 1, 2, 3
T_{10}, T_{20}	Feed stream temperatures to vessels 1, 2
F_1, F_2	Effluent flow rate from vessels 1, 2
F_{10}, F_{20}	Steady-state feed stream flow rates to vessels 1, 2
F_r, F_p	Flow rates of the recycle and purge
V_1, V_2, V_3	Volumes of vessels 1, 2, 3
E_1, E_2	Activation energy for reactions 1, 2
k_1, k_2	Pre-exponential values for reactions 1, 2
ΔH_1, ΔH_2	Heats of reaction for reactions 1, 2
α_A, α_B, α_C	Relative volatilities of A, B, C
Q_1, Q_2, Q_3	Heat inputs into vessels 1, 2, 3
C_p, R, ρ	Heat capacity, gas constant and solution density

Table 2. Noise Parameters

	σ_p	ϕ	θ_p		σ_p	ϕ	θ_p		σ_p	ϕ	θ_p
x_{A1}	1	0.7	0.25	x_{A2}	1	0.7	0.25	x_{A3}	1	0.7	0.25
x_{B1}	1	0.7	0.25	x_{B2}	1	0.7	0.25	x_{B3}	1	0.7	0.25
T_1	10	0.7	2.5	T_2	10	0.7	2.5	T_3	10	0.7	2.5

where $x^T = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9] = [x_{A1} - x_{A1s}\ x_{B1} - x_{B1s}\ T_1 - T_{1s}\ x_{A2} - x_{A2s}\ x_{B2} - x_{B2s}\ T_2 - T_{2s}\ x_{A3} - x_{A3s}\ x_{B3} - x_{B3s}\ T_3 - T_{3s}]$ is the state, $u_1^T = [u_{11}\ u_{12}\ u_{13}] = [Q_1 - Q_{1s}\ Q_2 - Q_{2s}\ Q_3 - Q_{3s}]$ and $u_2 = F_{20} - F_{20s}$ are the manipulated inputs which are subject to the constraints $|u_{1i}| \le 10^6\ KJ/h$ $(i = 1,2,3)$ and $|u_2| \le 3$ m^3/h, and $w = w_k$ is a time varying bounded noise. The process of Eq. 14 with the distributed LMPC control architecture is shown in Figure 5.

To illustrate the theoretical results, we first design the Lyapunov-based controller $u_1 = h(x)$ which can stabilize the closed-loop system as follows[2]:

$$h(x) = \begin{cases} -\frac{L_fV + \sqrt{L_fV^2 + L_{g1}V^4}}{L_{g1}V^2} L_{g1}V & \text{if} L_{g1}V \neq 0 \\ 0 & \text{if} L_{g1}V = 0 \end{cases} \tag{16}$$

where $L_fV = \frac{\partial V}{\partial x}|_x f(x)$ and $L_{g1}V = \frac{\partial V}{\partial x}|_x g_1(x)$ denote the Lie derivatives of the scalar function V with respect to the vector fields f and g_1, respectively. We consider a control Lyapunov function $V(x) = x^T Px$ with P being the following weight matrix

$$P = \text{diag}^{\S}(5.2 \times 10^{12}\left[4\ \ 4\ \ 10^{-4}\ \ 4\ \ 4\ \ 10^{-4}\ \ 4\ \ 4\ \ 10^{-4})\right]).$$

The values of the weights in P have been chosen in such a way that the Lyapunov-based controller of Eq. 16 satisfies the input constraints, stabilizes the closed-loop system and provides good closed-loop performance.

Based on the Lyapunov-based controller of Eq. 16, we design the centralized and the distributed LMPCs. In the simulations, the same parameters are used for both control designs. The prediction step is the same as the sampling time, that is $\Delta = 0.02$ h $= 1.2$ min; the prediction horizon is chosen to be $N = 6$; and the weight matrices for the LMPC designs are chosen as:

$$Q_c = \text{diag}(\left[2 \cdot 10^3\ 2 \cdot 10^3\ 2.5\ \ 2 \cdot 10^3\ 2 \cdot 10^3\ 2.5\ \ 2 \cdot 10^3\ 2 \cdot 10^3\ 2.5\right])$$

and $R_{c1} = \text{diag}([5{\cdot}10^{-12}\ 5{\cdot}10^{-12}\ 5{\cdot}10^{-12}])$ and $R_{c2} = 100$.

The state and input trajectories of process of Eq. 14 under the distributed LMPC design and the centralized LMPC design from the initial state:

$$x(0)^T = [0.890\ 0.110\ 388.7\ 0.886\ 0.113\ 386.3\ 0.748\ 0.251\ 390.6].$$

are shown in Figures 6 and 7. Figure 6 shows that both the distributed and the centralized LMPC designs give similar closed-loop performance and drive the temperatures and the mass fractions in the closed-loop system close to the desired steady-state in about 0.3 and 0.5 h, respectively.

§diag(v) denotes a matrix with its diagonal elements being the elements of vector v and all the other elements being zeros.

Table 3. Steady-State Values for u_{1s} and u_{2s}

Q_{1s}	12.6×10^5[KJ/h]	Q_{3s}	11.88×10^5[KJ/h]
Q_{2s}	13.32×10^5[KJ/h]	F_{20s}	$5.04 \times$ [m^3/h]

We have also carried out a set of simulations to compare the distributed LMPC design with the centralized LMPC design with the same parameters from a performance index point of view. Table 5 shows the total cost computed for 15 different closed-loop simulations under the distributed LMPC design and the centralized LMPC design. To carry out this comparison, we have computed the total cost of each simulation with different operating conditions (different initial states and process noise) based on the index of the following form

$$\sum_{i=0}^{M} x(t_i)^T Q_c x(t_i) + u_1(t_i)^T R_{c1} u_1(t_i) + u_2(t_i)^T R_{c2} u_2(t_i)$$

where t_0 is the initial time of the simulations and $t_M = 1$ h is the end of the simulations. As we can see in Table 5, the distributed LMPC design has a cost lower than the centralized LMPC design in 10 out of 15 simulations. This illustrates that in this example, the closed-loop performance of the distributed LMPC design is comparable to the one of the centralized LMPC design.

Remark 18. Table 5 shows that both controllers yield a similar performance for this particular process, but in general there is no guarantee that the total performance cost along the closed-loop system trajectories of either control scheme should be better than the other because the solution provided by the centralized LMPC and the proposed distributed LMPC are proved to be feasible and stabilizing but the convergence of the cost under distributed MPC to the one under centralized MPC is not established. This is because the communication between the two distributed MPCs is limited to one directional and moreover, the controllers are implemented in a receding horizon scheme and the prediction horizon is finite. In addition, there are disturbances modeled by stochastic noise in the simulations which introduce uncertainty in the results.

Moreover, we have studied the importance of communicating optimal input trajectories of LMPC 2 to LMPC 1. We have carried out a set of simulations in which both LMPC controllers operate in a decentralized manner; that is, LMPC 2 does not send its optimal input trajectory to LMPC 1 each sampling time (there is no communication between the two LMPCs). To evaluate its control input, LMPC 1 assumes that LMPC 2 applies the steady-state input F_{20s}; that is $u_2 = 0$. The same parameters as in previous simulations are used for the controllers. Figures 8 and 9 show the results under

Table 4. Steady-State Values for x_s

x_{A1s}	0.605	x_{A2s}	0.605	x_{A3s}	0.346
x_{B1s}	0.386	x_{B2s}	0.386	x_{B3s}	0.630
T_{1s}	425.9[K]	T_{2s}	422.6[K]	T_{3s}	427.3[K]

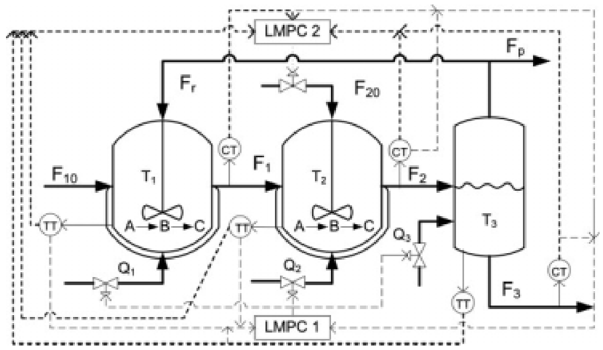

Figure 5. Reactor-separator process with distributed LMPC control architecture.

the decentralized LMPC design. From Figure 8, we can see that for this particular example, this control scheme can not stabilize the system at the required steady-state. This result is expected because when there is no communication between the two distributed controllers, they can not coordinate their control actions and each controller views the input of the other controller as a disturbance that has to be rejected.

We have also carried out a set of simulations to compare the computation time needed to evaluate the distributed LMPC with that of the centralized LMPC. The simulations have been carried out using Matlab in a Pentium 3.20 GHz. The optimization problems have been solved using the built-in function *fmincom* of Matlab. To solve the ODE model of Eq. 14, an Euler method with a fixed integration time of 0.001 h has been implemented in a mex DLL using C programming language. For 50 evaluations, the mean time to solve the centralized LMPC is 9.40 s; the mean times to solve LMPC 1 and LMPC 2 are 3.19 and 4.53 s, respectively. From this set of simulations, we see that the computation time needed to solve the centralized LMPC is larger than the sum of the values for LMPC 1 and LMPC 2 even though the closed-loop performance in terms of the total performance cost is comparable to the one of the distributed LMPC method. This is because the centralized LMPC has to optimize both the inputs u_1 and u_2 in one optimization problem and the distributed LMPC has to solve two smaller (in terms of decision variables) optimization problems.

Following Remark 10, we have also carried out a set of simulations to illustrate that the optimization processes of LMPC 1 and LMPC 2 can be terminated at any time to get sub-optimal solutions without loss of the closed-loop stability. In this set of simulations, we assume that the allowable evaluation times of LMPC 1 and LMPC 2 at each sampling time are 1 and 2, and we terminate the optimization processes of LMPC 1 and LMPC 2 when they have been carried out for 1 and 2, respectively. The closed-loop state and input trajectories under the distributed LMPC design with limited and unconstrained computation time are shown in Figures 10 and 11. From Figure 10, we see that the distributed LMPC design with limited evaluation time can stabilize the closed-

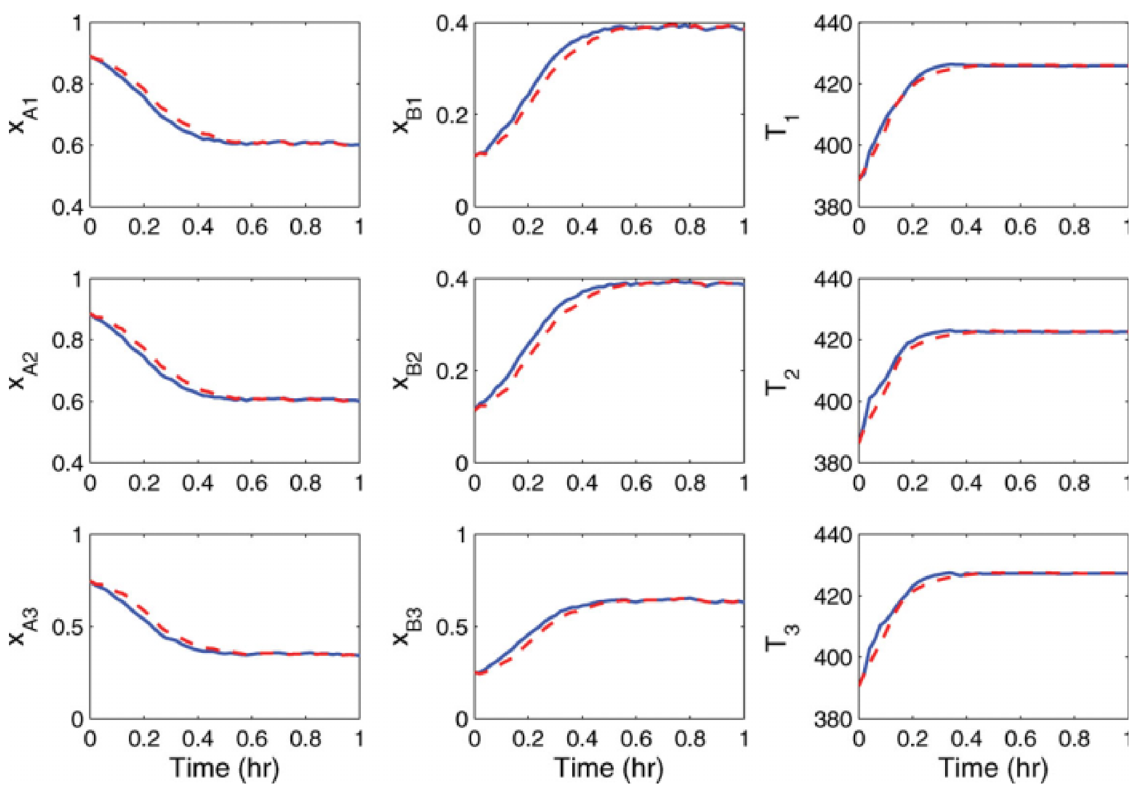

Figure 6. State trajectories of the process of Eq. 14 under the distributed LMPC design (solid lines) and centralized LMPC design (dashed lines).

[Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

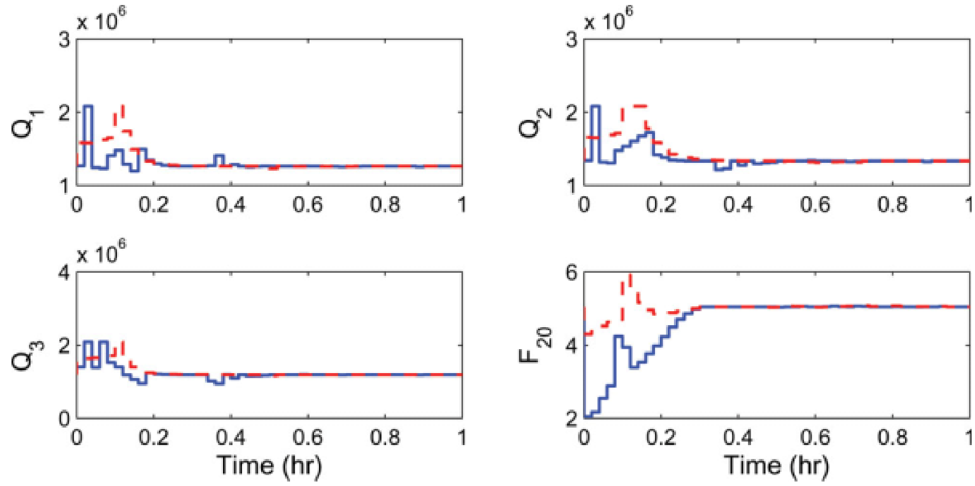

Figure 7. Input trajectories of the process of Eq. 14 under the distributed LMPC design (solid lines) and centralized LMPC design (dashed lines).

[Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Table 5. Total Performance Cost Along the Closed-Loop System Trajectories

Sim.	Distr.	Centr.	sim.	Distr.	Centr.	sim.	Distr.	Centr.
1	65216	70868	6	83776	66637	11	62714	70951
2	70772	73112	7	61360	68897	12	76348	70547
3	57861	67723	8	47070	66818	13	49914	66869
4	62396	70914	9	79658	64342	14	89059	72431
5	60407	67109	10	65735	72819	15	78197	70257

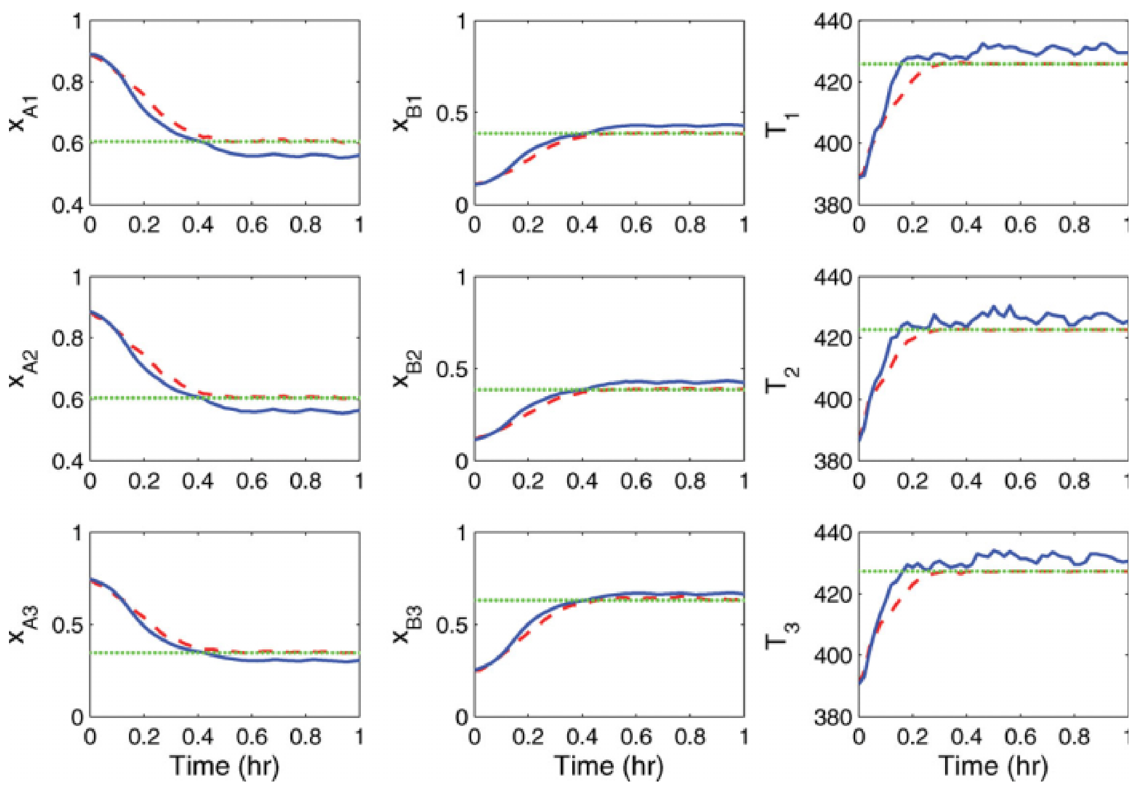

Figure 8. State trajectories of the process of Eq. 14 under the decentralized LMPC design (solid) and distributed LMPC design (dashed).

[Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

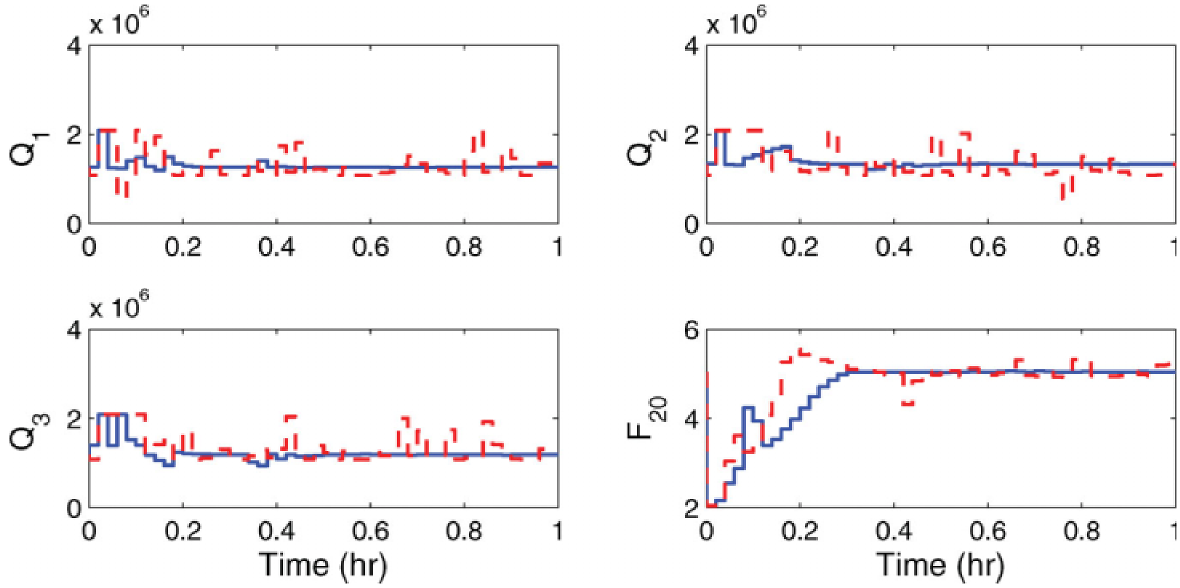

Figure 9. Input trajectories of system Eq. 14 under the decentralized LMPC design (solid) and distributed LMPC design (dashed).

[Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

loop system but the state responses are slower, leading to a higher cost (57778) compared with the one (47117) obtained under the distributed LMPC design with unconstrained computation time.

Conclusion

Currently, process control systems utilize dedicated, wired control networks to achieve key closed-loop properties like

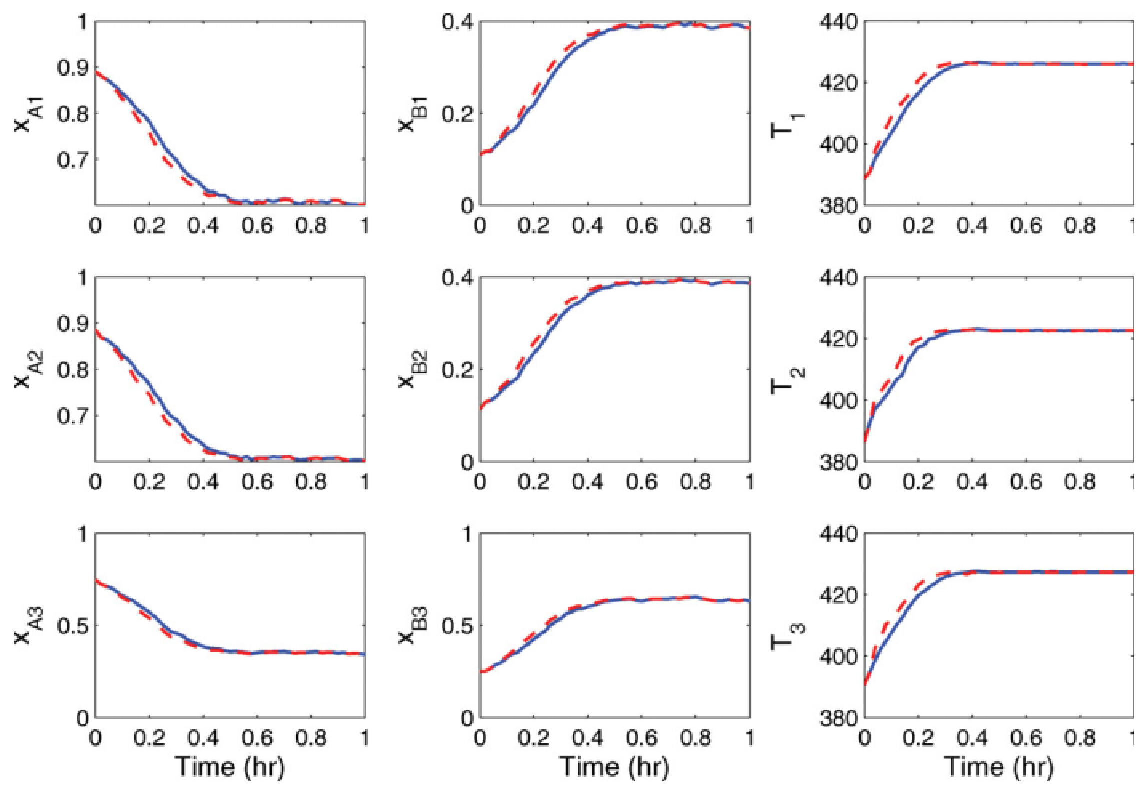

Figure 10. State trajectories of the process of Eq. 14 under the distributed LMPC design with limited (solid lines) and unconstrained (dashed lines) evaluation time.

[Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

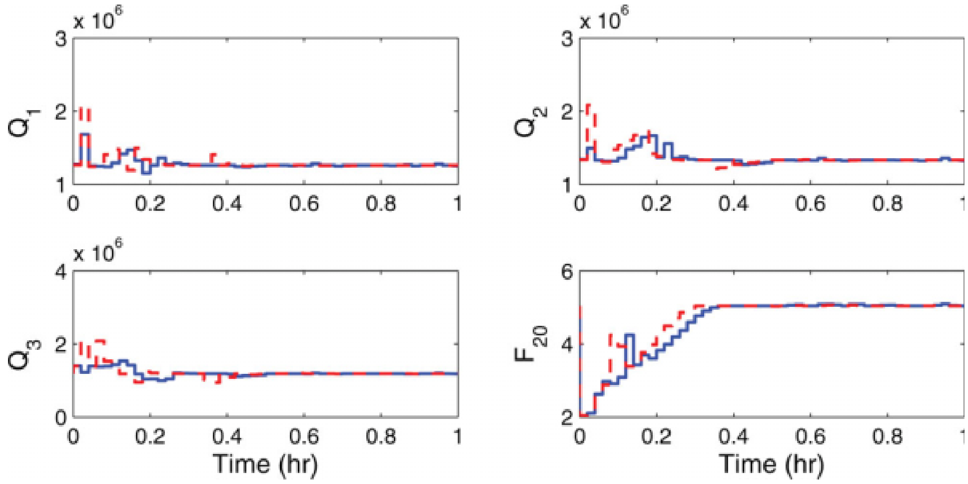

Figure 11. Input trajectories of the process of Eq. 14 under the distributed LMPC design with limited (solid lines) and unconstrained (dashed lines) evaluation time.

stability, set-point tracking, and robustness to disturbances. With the advent of sensor and actuator network (both wired and wireless) technology, however, there is a growing realization that low-cost sensor and actuator networks can play an important auxiliary role to the existing control systems by collecting and transferring additional data to the control system and by utilizing additional control actuators. Chemical plant operation could substantially benefit from an efficient integration of the existing, point-to-point control networks with additional networked actuator/sensor devices. Motivated by these considerations, in the present work, we focused on a class of nonlinear control problems that arise when new control systems which may use networked sensors and/or actuators are added to already operating control loops to improve closed-loop performance. To address this control problem, a distributed model predictive control method was introduced where both the pre-existing control system and the new control system are designed via Lyapunov-based model predictive control theory. The proposed distributed model predictive control design preserves the stability properties of the Lyapunov-based controller, improves the closed-loop performance, and allows handling input constraints. In addition, the proposed distributed control design requires reduced communication between the two distributed controllers since it requires that these controllers communicate only once at each sampling time and is computationally more efficient compared to the corresponding centralized model predictive control design. Extensive simulations using a benchmark chemical plant example, described by a nonlinear model, demonstrated the applicability and effectiveness of the proposed control method.

Our future work includes to extend the distributed MPC scheme to systems with asynchronous measurements and communication data losses and to generalize the proposed design approach to include multiple distributed MPC controllers.

Acknowledgements

Financial support from NSF and the European Commission, INFSOICT-223866, is gratefully acknowledged.

Literature Cited

1. Nimmo I. Adequately address abnormal operations. *Chem Eng Prog*. 1995;91:36–45.
2. Ydstie EB. New vistas for process control: integrating physics and communication networks. *AIChE J*. 2002;48:422–426.
3. Davis JF. *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), 2007.
4. Neumann P. Communication in industrial automation: what is going on? *Control Eng Practice*. 2007;15:1332–1347.
5. Christofides PD, Davis JF, El-Farra NH, Clark D, Harris KRD, Gipson JN. Smart plant operations: vision, progress and challenges. *AIChE J*. 2007;53:2734–2741.
6. Davis JF, Edgar TF. *Report from NSF Roadmap Development Workshop on Zero-Incident, Zero-Emission Smart Manufacturing* (see http://www.oit.ucla.edu/nsf-evo-2008/program/SPM-Workshop-Report.pdf for the pdf file of this report), 2008.
7. Liu J, Muñoz de la Peña D, Ohran B, Christofides PD, Davis JF. A two-tier architecture for networked process control. *Chem Eng Sci*. 2008;63:5394–5409.
8. Sun Y, El-Farra NH. Quasi-decentralized model-based networked control of process systems. *Comput Chem Eng*. 2008;32:2016–2029.
9. Camponogara E, Jia D, Krogh BH, Talukdar S. Distributed model predictive control. *IEEE Control Syst Mag*. 2002;22:44–52.
10. Rawlings JB, Stewart BT. *Coordinating multiple optimization-based controllers: new opportunities and chanllenges*. In: *Procedings of 8th IFAC Symposium on Dynamics and Control of Process*, Vol. 1, Cancun, Mexico, 2007:19–28.
11. Dunbar WB. Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Trans Automatic Control*. 2007;52:1249–1263.
12. Richards A, How JP. Robust distributed model predictive control. *Int J Control*. 2007;80:1517–1531.
13. Jia D, Krogh B. Min-max feedback model predictive control for distributed control with communication. In: Proceedings of the American Control Conference. Anchorage, 2002:4507–4512.
14. Venkat AN, Rawlings JB, Wright SJ. Stability and optimality of distributed model predictive control. In: Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference ECC 2005, Seville, Spain, 2005:6680–6685.
15. Keviczky T, Borrelli F, Balas GJ. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*. 2006;42:2105–2115.
16. Magni L, Scattolini R. Stabilizing decentralized model predictive control of nonlinear systems. *Automatica*. 2006;42:1231–1236.

17. Raimondo DM, Magni L, Scattolini R. Decentralized mpc of nonlinear system: an input-to-state stability approach. *Int J Robust Nonlinear Control*. 2007;17:1651–1667.
18. Kokotovic P, Arcak M. Constructive nonlinear control: a historical perspective. *Automatica*. 2001;37:637–662.
19. Christofides PD, El-Farra NH. *Control of Nonlinear and Hybrid Process Systems: Designs for Uncertainty, Constraints and Time-Delays*. Berlin, Germany: Springer-Verlag, 2005.
20. Lin Y, Sontag ED. A universal formula for stabilization with bounded controls. *Syst Control Lett*. 1991;16:393–397.
21. Sontag E. A 'universal' construction of Artstein's theorem on nonlinear stabilization. *Syst Control Lett*. 1989;13:117–123.
22. El-Farra NH, Christofides PD. Integrating robustness, optimality and constraints in control of nonlinear processes. *Chem Eng Sci*. 2001;56:1841–1868.
23. El-Farra NH, Christofides PD. Bounded robust control of constrained multivariabe nonlinear processes. *Chem Eng Sci*. 2003;58: 3025–3047.
24. Mhaskar P, El-Farra NH, Christofides PD. Predictive control of switched nonlinear systems with scheduled mode transitions. *IEEE Trans Automatic Control*. 2005;50:1670–1680.
25. Mhaskar P, El-Farra NH, Christofides PD Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control. *Syst Control Lett*. 2006;55:650–659.
26. Clarke F, Ledyaev Y, Sontag E. Asymtotic controllability implies feedback stabilization. *IEEE Trans Automatic Control*. 1997;42: 1394–1407.
27. Nešić D, Teel A, Kokotovic P. Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete time approximations. *Syst Control Lett*. 1999;38:259–270.
28. Tabuada P, Wang X. Preliminary results on state-trigered scheduling of stabilizing control tasks. In: *Proceedings of IEEE Conference on Decision and Control*, San Diego, CA. 2006:282–287.

Manuscript received Sept. 25, 2008, and final revision received Dec. 15, 2008.