# Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. Part I: Theory

**Conference Paper** *in* Proceedings of the American Control Conference · August 2010

DOI: 10.1109/ACC.2010.5531017 · Source: IEEE Xplore



Jinfeng Liu
University of Alberta


Panagiotis Christofides
University of California, Los Angeles

AIChE

# Sequential and Iterative Architectures for Distributed Model Predictive Control of Nonlinear Process Systems

**Jinfeng Liu and Xianzhong Chen**
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, Sevilla, 41092, Spain

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

Dept. of Electrical Engineering, University of California, Los Angeles, CA 90095

*DOI 10.1002/aic.12155*
*Published online January 22, 2010 in Wiley InterScience (www.interscience.wiley.com).*

*In this work, we focus on distributed model predictive control of large scale nonlinear process systems in which several distinct sets of manipulated inputs are used to regulate the process. For each set of manipulated inputs, a different model predictive controller is used to compute the control actions, which is able to communicate with the rest of the controllers in making its decisions. Under the assumption that feedback of the state of the process is available to all the distributed controllers at each sampling time and a model of the plant is available, we propose two different distributed model predictive control architectures. In the first architecture, the distributed controllers use a one-directional communication strategy, are evaluated in sequence and each controller is evaluated only once at each sampling time; in the second architecture, the distributed controllers utilize a bi-directional communication strategy, are evaluated in parallel and iterate to improve closed-loop performance. In the design of the distributed model predictive controllers, Lyapunov-based model predictive control techniques are used. To ensure the stability of the closed-loop system, each model predictive controller in both architectures incorporates a stability constraint which is based on a suitable Lyapunov-based controller. We prove that the proposed distributed model predictive control architectures enforce practical stability in the closed-loop system and optimal performance. The theoretical results are illustrated through a catalytic alkylation of benzene process example.*  *AIChE J*, 56: 2137–2149, 2010
*Keywords: model predictive control, distributed control, distributed optimization, large-scale systems, process control*

## Introduction

Model predictive control (MPC) is a popular control strategy based on using a model of the process to predict at each sampling time the future evolution of the system from the

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

current state along a given prediction horizon. Using these predictions, the manipulated input trajectory that minimizes a given performance index is computed solving a suitable optimization problem. To obtain finite dimensional optimization problems, MPC optimizes over a family of piecewise constant trajectories with a fixed sampling time and a finite prediction horizon. Once the optimization problem is solved, only the first manipulated input value is implemented, discarding the rest of the trajectory and repeating the optimization in the next sampling step.[1,2] 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. When the number of the state variable and manipulated inputs of the process, however, becomes large, the computational burden of the centralized optimization problem may increase significantly and may impede the applicability of a centralized MPC, especially in the case where nonlinear process models are used in the MPC. One feasible alternative to overcome this problem is to utilize a distributed MPC architecture in which the manipulated inputs are computed by more than one optimization problems in a coordinated fashion. The objective of the present study is to propose two distributed MPC architectures for nonlinear process systems to reduce the computational burden of computing the values of the manipulated inputs and to coordinate the distributed MPC controllers in a suitable fashion to achieve stability and optimal performance of the closed-loop system.

With respect to available results on distributed MPC architectures, 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 References 3–5 for reviews of results in this area. More specifically, in Reference 6, the problem of distributed control of dynamically coupled nonlinear systems that are subject to decoupled constraints was considered. In References 7, 8, the effect of the coupling was modeled as a bounded disturbance compensated using a robust MPC formulation. In Reference 9, 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 Reference 10, distributed MPC of decoupled systems (a class of systems of relevance in the context of multi-agents systems) was studied. In Reference 11, an MPC algorithm was proposed under the main condition that the system is nonlinear, discrete-time, and no information is exchanged between local controllers, and in Reference 12, MPC for nonlinear systems was studied from an input-to-state stability point of view. In Reference 13, a game theory based distributed MPC scheme for linear systems coupled through the inputs was proposed. In a recent study,[14] we proposed a distributed MPC architecture with one-directional communication for general nonlinear process systems. In this architecture, two separate MPC controllers designed via Lyapunov-based MPC (LMPC) were considered, in which one LMPC was used to guarantee the stability of the closed-loop system and the other LMPC was used to improve the closed-loop performance. Generally, the computational burden of these distributed MPC methods is smaller compared to the one of the corresponding centralized MPC because of the formulation of optimization problems with a smaller number of decision variables.

In this work, we focus on distributed MPC of large scale nonlinear process systems in which several distinct sets of manipulated inputs are used to regulate the process. For each set of manipulated inputs, a different MPC is used to compute the control actions, which is able to communicate with the rest of the controllers in making its decisions. Under the assumption that feedback of the state of the process is available to all the distributed controllers at each sampling time and a model of the plant is available, we propose two different distributed MPC architectures designed via LMPC techniques. In the first architecture, the distributed controllers use a one-directional communication strategy, are evaluated in sequence and each controller is evaluated only once at each sampling time; in the second architecture, the distributed controllers utilize a bi-directional communication strategy, are evaluated in parallel and iterate to improve closed-loop performance. To ensure the stability of the closed-loop system, each model predictive controller in both architectures incorporates a stability constraint which is based on a suitable Lyapunov-based controller. We prove that the proposed distributed MPC architectures enforce practical stability in the closed-loop system and optimal performance. The theoretical results are illustrated through a catalytic alkylation of benzene process example.

## Preliminaries

### *Problem formulation*

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

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

where $x(t) \in R^{n_x}$ denotes the vector of process state variables, $u_i(t) \in R^{m_{ui}}$, $i = 1,\ldots,m$, are $m$ sets of control (manipulated) inputs and $w(t) \in R^{n_w}$ denotes the vector of disturbance variables. The $m$ sets of inputs are restricted to be in $m$ nonempty convex sets $U_i \subseteq R^{m_{ui}}$, $i = 1,\ldots,m$, which are defined as follows:

$$U_i := \{u_i \in R^{m_{u_i}} : |u_i| \leq u_i^{\max}\}^*, i = 1, \ldots, m$$

where $u_i^{\max}$, $i = 1,\ldots,m$, are the magnitudes of the input constraints. The disturbance vector is bounded, i.e., $w(t) \in W$ where

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

We assume that $f$, $g_i$, $i = 1,\ldots,m$, and $k$ are locally Lipschitz vector functions and that the origin is an equilibrium of the unforced nominal system (i.e., system of Eq. 1 with $u_i(t) = 0$, $i = 1,\ldots,m$, $w(t) = 0$ for all $t$) which implies that $f(0) = 0$. We also assume that the state $x$ of the system is sampled synchronously and the time instants at which we have state 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 $\Delta$ is the sampling time.

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

**Remark 1.** In general, distributed control systems are formulated based on the assumption that the controlled process systems consist of decoupled or partially coupled subsystems. However, we consider a fully coupled process model; this is a very common occurrence in chemical process control as we will illustrate in the Application to a Chemical Process Example section. In our future study, we will extend the proposed distributed control systems to the case in which only local state information is available to each distributed controller based on distributed state estimation.

**Remark 2.** Note that the assumption that $f$, $g_i$, $i = 1,\ldots,m$ and $k$ are locally Lipschitz vector functions is a reasonable assumption for most of chemical processes. Note also that the assumption that the state $x$ of the system is sampled synchronously is a widely used assumption in the research of process control. The proposed control system designs can be extended to the case where only part of the state $x$ is measurable by designing an observer to estimate the whole state vector from output measurements and by designing the control system based on the measured and estimated states. In this case, the stability properties of the resulting output feedback control systems are affected by the convergency of the observer and need to be carefully studied.

### *Lyapunov-based controller*

We assume that there exists a Lyapunov-based controller $h(x) = [h_1(x) \ldots h_m(x)]^T$ with $u_i = h_i(x)$, $i = 1,\ldots,m$, which renders the origin of the nominal closed-loop system asymptotically stable while satisfying the input constraints for all the states $x$ inside a given stability region. We note that this assumption is essentially equivalent to the assumption that the process is stabilizable or that the pair $(A, B)$ in the case of linear systems is stabilizable. Using converse Lyapunov theorems,[15–17] this assumption implies that there exist functions $\alpha_i(\cdot)$, $i = 1,2,3,4$ of class $\mathcal{K}^\dagger$ and a continuously differentiable Lyapunov function $V(x)$ for the nominal closed-loop system which is continuous and bounded in $R^{n_x}$, that satisfy the following inequalities:

$$\begin{aligned}
&\alpha_1(|x|) \le V(x) \le \alpha_2(|x|),\\
&\frac{\partial V(x)}{\partial x}\left(f(x) + \sum_{i=1}^{m} g_i(x)h_i(x)\right) \le -\alpha_3(|x|),\\
&\left|\frac{\partial V(x)}{\partial x}\right| \le \alpha_4(|x|),\\
&h_i(x) \in U_i,\ i = 1,\ldots,m
\end{aligned} \tag{2}$$

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 Lyapunov-based controller $h(x)$. The construction of $V(x)$ can be carried out in a number of ways using systematic techniques like, for example, sum-of-squares methods.

By continuity, the local Lipschitz property assumed for the vector fields $f$, $g_i$, $i = 1,\ldots,m$, and $k$ and taking into account that the manipulated inputs $u_i$, $i = 1,\ldots,m$, and the disturbance $w$ are bounded in convex sets, there exists a positive constant $M$ such that

$$\left|f(x) + \sum_{i=1}^{m} g_i(x)u_i + k(x)w\right| \le M \tag{3}$$

for all $x \in \Omega_\rho$, $u_i \in U_i$, $i = 1,\ldots,m$, and $w \in W$. In addition, by the continuous differentiable property of the Lyapunov function $V(x)$ and the Lipschitz property assumed for the vector field $f$, there exist positive constants $L_x$, $L_{u_i}$, $i = 1,\ldots,m$, and $L_w$ such that

$$\begin{aligned}
&\left|\frac{\partial V}{\partial x}f(x) - \frac{\partial V}{\partial x}f(x')\right| \le L_x|x - x'|\\
&\left|\frac{\partial V}{\partial x}g_i(x) - \frac{\partial V}{\partial x}g_i(x')\right| \le L_{u_i}|x - x'|,\ i = 1,\ldots,m\\
&\left|\frac{\partial V}{\partial x}k(x)\right| \le L_w
\end{aligned} \tag{4}$$

for all $x$, $x' \in \Omega_\rho$, $u_i \in U_i$, $i = 1,\ldots,m$, and $w \in W$.

**Remark 3.** Different state feedback control laws for nonlinear systems have been developed using Lyapunov techniques; the reader may refer to References 17–21 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.

### *Centralized LMPC*

To take advantage of all the sets of manipulated inputs, one option is to design a centralized MPC controller. To guarantee robust stability of the closed-loop system, the MPC controller must include a set of stability constraints. To do this, we propose to use the LMPC controller proposed in References 22 and 23, which guarantees practical stability of the closed-loop system, allows for an explicit characterization of the stability region, and yields a reduced complexity optimization problem. LMPC is based on uniting receding horizon control with control Lyapunov functions and computes the manipulated input trajectory solving a finite horizon constrained optimal control problem. The LMPC controller is based on the Lyapunov-based controller $h(x)$. The controller $h(x)$ is used to define a stability constraint for the LMPC controller which guarantees that the LMPC controller inherits the stability and robustness properties of the Lyapunov-based controller $h(x)$. The LMPC controller introduced in References 22 and 23 is based on the following optimization problem:

$$\min_{u_{c1}\ldots u_{cm}\in S(\Delta)} \int_0^{N\Delta} [\tilde{x}^T(\tau)Q_c\tilde{x}(\tau) + \sum_{i=1}^{m} u_{ci}^T(\tau)R_{ci}u_{ci}(\tau)]d\tau \tag{5a}$$

$$\text{s.t. } \dot{\tilde{x}}(\tau) = f(\tilde{x}(\tau)) + \sum_{i=1}^{m} g_i(\tilde{x}(\tau))u_{ci} \tag{5b}$$

$$u_{ci}(\tau) \in U_i,\ i = 1,\ldots,m \tag{5c}$$

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

$$\sum_{i=1}^{m} \frac{\partial V(x)}{\partial x}g_i(x(t_k))u_{ci}(0) \le \sum_{i=1}^{m} \frac{\partial V(x)}{\partial x}g_i(x(t_k))h_i(x(t_k)) \tag{5e}$$

where $S(\Delta)$ is the family of piece-wise constant functions with sampling period $\Delta$, $N$ is the prediction horizon, $Q_c$ and $R_{ci}$, $i =$

†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 $\Omega_\rho$ to denote the set $\Omega_\rho := \{x \in R^{n_x} | V(x) \le \rho\}$.

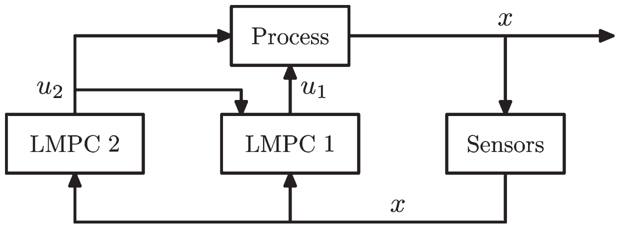


**Figure 1. Distributed MPC scheme proposed in Reference 14.**

1,…, $m$, 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 with $u_i$, $i = 1,\ldots,m$, the input trajectory computed by the LMPC of Eq. 5.

The optimal solution to this optimization problem is denoted by $u^*_{ci}(\tau|t_k)$, $i = 1,\ldots,m$, which is defined for $\tau \in [0,N\Delta)$. The LMPC 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 $u^*_{ci}(t - t_k|t_k)$, $i = 1,\ldots,m$ are applied to the closed-loop system for $t \in [t_k,t_{k+1})$.

The optimization problem of Eq. 5 does not depend on the uncertainty and guarantees that the system in closed-loop with the LMPC of Eq. 5 maintains the stability properties of the Lyapunov-based controller. The constraint of Eq. 5e guarantees that the value of the time derivative of the Lyapunov function at the initial evaluation time of the centralized LMPC controller is lower or equal to the value obtained if only the Lyapunov-based controller $h(x)$ is implemented in the closed-loop system in a sample-and-hold fashion. This is the constraint that allows proving that the centralized LMPC controller inherits the stability and robustness properties of the Lyapunov-based controller.

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

$$u_i(t) = u^*_{ci}(t - t_k|t_k),\ i = 1,\ldots,m,\ \forall t \in [t_k, t_{k+1}). \qquad (6)$$

In what follows, we refer to this controller as the centralized LMPC. 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 $\Omega_\rho$ for a sufficient small sampling time $\Delta$ and disturbance upper bound $\theta$. This property is also guaranteed by the Lyapunov-based controller $h(x)$ when it is implemented in a sample-and-hold fashion (see Refs. 24 and 25 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[23]) in the computation of the control actions within an online optimization framework to improve closed-loop performance.

## Distributed MPC Architectures

In our previous study,[14] we introduced a distributed MPC architecture for nonlinear process systems based on the scheme shown in Figure 1. In this distributed MPC architecture, two MPC controllers designed via LMPC were considered. One of the two LMPC controllers (LMPC 1) was designed to guarantee the stability of the closed-loop system and the other LMPC controller (LMPC 2) was designed to improve the closed-loop performance while maintaining the closed-loop stability achieved by LMPC 1. This distributed MPC architecture required one-directional communication between the two distributed controllers and was proved that it guarantees practical stability of the closed-loop system and has the potential to maintain the closed-loop stability and performance in the face of new or failing controllers or actuators (for example, a zero input of LMPC 2 does not affect the closed-loop stability) and to reduce computational burden in the evaluation of the optimal manipulated inputs compared with a fully centralized LMPC controller of the same input/output-space dimension.

In the present study, our objective is to extend our results in Reference 14 and propose distributed MPC architectures including multiple MPCs for large scale nonlinear process systems. Specifically, we propose two different distributed MPC architectures. The first distributed MPC architecture is a direct extension of our previous study in Reference 14 in which different MPC controllers are evaluated in sequence, only once at each sampling time and require only one-directional communication between consecutive distributed controllers (i.e., the distributed controllers are connected by pairs). In the second architecture, different MPC controllers are evaluated in parallel, once or more than once at each sampling time depending on the number of iterations, and bi-directional communication among all the distributed controllers (i.e., the distributed controllers are all interconnected) is used.

In each proposed architecture, we will design $m$ LMPC controllers to compute $u_i$, $i = 1,\ldots,m$, and refer to the LMPC controller computing the input trajectories of $u_i$ as LMPC $i$.

### *Sequential distributed LMPC*

In this subsection, we will discuss the direct extension of the results in Reference 14 to include multiple LMPC controllers, in which different LMPC controllers are evaluated in sequence, once at each sampling time and one-directional communication between consecutive distributed controllers (i.e., the distributed controllers are connected by pairs) is used. A schematic of this architecture is shown in Figure 2. We first present the proposed implementation strategy of this

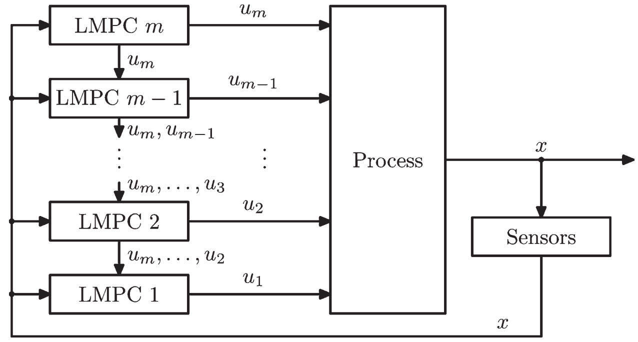


**Figure 2. Sequential distributed LMPC architecture.**

distributed MPC architecture and then design the corresponding LMPC controllers. The proposed implementation strategy of this distributed MPC architecture is as follows:

1. At each sampling time $t_k$, all the LMPC controllers 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 $u_i$, $i = m,\ldots,j$, to LMPC $j-1$.

In this architecture, each LMPC controller only sends its future input trajectory and the future input trajectories it received to the next LMPC controller (i.e., LMPC $j$ sends input trajectories to LMPC $j-1$). This implies that LMPC $j$, $j = m,\ldots,2$, does not have any information about the values that $u_i$, $i = j-1,\ldots,1$ will take when the optimization problems of the LMPC controllers are designed. To make a decision, LMPC $j$, $j = m,\ldots,2$ must assume trajectories for $u_i$, $i = j-1,\ldots,1$, along the prediction horizon. To this end, the Lyapunov-based controller $h(x)$ is used. To inherit the stability properties of the controller $h(x)$ (i.e., the stability properties of $h(x)$), each control input $u_i$, $i = 1,\ldots,m$ must satisfy a constraint that guarantees a given minimum contribution to the decrease rate of the Lyapunov function $V(x)$. Specifically, the proposed design of the LMPC $j$, $j = 1,\ldots,m$, is based on the following optimization problem:

$$u^*_{s,j}(\tau|t_k) = \arg\min_{u_{s,j}\in S(\Delta)} \int_0^{N\Delta} [\tilde{x}^T(\tau)Q_c\tilde{x}(\tau) + \sum_{i=1}^{m} u_{s,i}(\tau)^T R_{ci} u_{s,i}(\tau)]d\tau \quad (7a)$$

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

$$u_{s,i}(\tau) = h_i(\tilde{x}(j\Delta)),\ i = 1,\ldots,j-1,\ \forall\,\tau \in [l\Delta,(l+1)\Delta),\ l = 0,\ldots,N-1 \quad (7c)$$

$$u_{s,i}(\tau) = u^*_{s,i}(\tau|t_k), i = j+1,\ldots,m \quad (7d)$$

$$u_{s,j}(\tau) \in U_j \quad (7e)$$

$$\tilde{x}(0) = x(t_k) \quad (7f)$$

$$\frac{\partial V(x)}{\partial x} g_j(x(t_k))u_{s,j}(0) \le \frac{\partial V(x)}{\partial x} g_j(x(t_k))h_j(x(t_k)) \quad (7g)$$

where $\tilde{x}$ is the predicted trajectory of the nominal system with $u_i$, $i = j+1,\ldots,m$, the input trajectory computed by the LMPC controllers of Eq. 7 evaluated before LMPC $j$, $u_i$, $i = 1,\ldots,j-1$, the corresponding elements of $h(x)$ applied in a sample-and-hold fashion and $u^*_{s,i}(\tau|t_k)$ denotes the future input trajectory of $u_i$ obtained by LMPC $i$ of the form of Eq. 7. The optimal solution to the optimization problem of Eq. 7 is denoted $u^*_{s,j}(\tau|t_k)$, which is defined for $\tau \in [0,N\Delta)$.

The constraint of Eq. 7b is the nominal model of the system of Eq. 1, which is used to predict the future evolution of the system; the constraints of Eq. 7c define the value of the inputs evaluated after $u_j$ (i.e., $u_i$ with $i = 1,\ldots,j-1$); the constraints of Eq. 7d define the value of the inputs evaluated before $u_j$ (i.e., $u_i$ with $i = j+1,\ldots,m$); the constraint of Eq. 7e is the constraint on the manipulated input $u_j$; the constraint of Eq. 7f sets the initial state for the optimization problem; the constraint of Eq. 7g guarantees that the contribution of input $u_j$ to the decrease rate of the time derivative of the Lyapunov function at the initial evaluation time, if $u_j = u^*_{s,j}(0|t_k)$ is applied, is bigger 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 the proposed controller.

The manipulated inputs of the proposed control design of Eq. 7 are defined as follows:

$$u_i(t) = u^*_{s,i}(t-t_k|t_k),\ i = 1,\ldots,m,\ \forall t \in [t_k,t_{k+1}). \quad (8)$$

In what follows, we refer to this distributed LMPC architecture as the sequential distributed LMPC.

**Remark 4.** Note that, to simplify the description of the implementation strategy proposed above in this subsection, we do not distinguish LMPC $m$ and LMPC 1 from the others. We note that LMPC $m$ does not receive any information from the other controllers and LMPC 1 does not have to send information to any other controller.

The proposed distributed LMPC architecture of Eqs. 7 and 8 computes the inputs $u_i$, $i = 1,\ldots,m$, applied to the system of Eq. 1 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 constraint of Eq. 7g. This property is presented in Theorem 1 below.

**Theorem 1.** *Consider the system of* Eq. 1 *in closed-loop under the distributed LMPC of Eqs.* 7 *and* 8 *based on the controller $h(x)$ that satisfies the condition of* Eq. 2 *with class $\mathcal{K}$ functions $\alpha_i(\cdot)$, $i = 1,2,3,4$. Let $\varepsilon_w > 0$, $\Delta > 0$ and $\rho > \rho_s > 0$ satisfy the following constraint*:

$$-\alpha_3(\alpha_2^{-1}(\rho_s)) + L^* \le -\varepsilon_w/\Delta \quad (9)$$

where $L^* = (L_x + \sum_{i=1}^{m} L_{u_i} u_i^{\max})M + L_w\theta$ with $M$, $L_x$, $L_{u_i}$ ($i = 1,\ldots,m$) and $L_w$ being defined in Eqs. 3 and 4. For any $N \ge 1$, if $x(t_0) \in \Omega_\rho$ and if $\rho^* \le \rho$ where

$$\rho^* = \max\{V(x(t+\Delta)) : V(x(t)) \le \rho_s\}, \quad (10)$$

then the state $x(t)$ of the closed-loop system is ultimately bounded in $\Omega_{\rho^*}$.

**Proof.** The proof consists of two parts. We first prove that the optimization problem of Eq. 7 is feasible for all $j = 1,\ldots,m$ and $x \in \Omega_\rho$. Then we prove that, under the proposed distributed LMPC of Eqs. 7 and 8, the state of the system of Eq. 1 is ultimately bounded in $\Omega_{\rho^*}$. Note that the constraint of Eq. 7g of each distributed controller is independent from the decisions that the rest of the distributed controllers make.

**Part 1:** To prove the feasibility of the optimization problem of Eq. 7, we only have to prove that there exists a $u_{s,j}(0)$ which satisfies the input constraint of Eq. 7e and the constraint of Eq. 7g. This is because the constraint of Eq. 7g is only enforced on the first prediction step of $u_{s,j}(\tau)$ and in the prediction time $\tau \in [\Delta, N\Delta)$, the input constraint of Eq. 8 can be easily satisfied with $u_{s,j}(\tau)$ being any value in the convex set $U_j$.

We assume that $x(t_k) \in \Omega_\rho$ ( $x(t)$ is bounded in $\Omega_\rho$ which will be proved in Part 2). It is easy to verify that the value of $u_{s,j}$ such that $u_{s,j}(0) = h_j(x(t_k))$ satisfies the input constraint of Eq. 7e (assumed property of $h(x)$ for $x \in \Omega_\rho$) and the constraint of Eq. 7g, thus, the feasibility of the optimization problem of LMPC $j$, $j = 1,\ldots, m$, is guaranteed.

**Part 2:** From the condition of Eq. 2 and the constraint of Eq. 7g, if $x(t_k) \in \Omega_\rho$, it follows that

$$\begin{aligned}\frac{\partial V}{\partial x}&\left(f(x(t_k)) + \sum_{i=1}^{m} g_i(x(t_k))u^*_{s,i}(0|t_k)\right)\\ &\leq \frac{\partial V}{\partial x}\left(f(x(t_k)) + \sum_{i=1}^{m} g_i(x(t_k))h_i(x(t_k)\right)\\ &\leq -\alpha_3(|x(t_k)|).\end{aligned} \tag{11}$$

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

$$\dot{V}(x(t)) = \frac{\partial V}{\partial x}\left(f(x(t)) + \sum_{i=1}^{m} g_i(x(t))u^*_{s,i}(0|t_k) + k(x(t))w(t)\right). \tag{12}$$

Adding and subtracting $\frac{\partial V}{\partial x}(f(x(t_k)) + \sum_{i=1}^{m} g_i(x(t_k))u^*_{s,i}(0|t_k))$ and taking into account Eq. 11, we obtain the following inequality

$$\begin{aligned}\dot{V}(x(t)) \leq& -\alpha_3(|x(t_k)|)\\ &+\frac{\partial V}{\partial x}\left(f(x(t)) + \sum_{i=1}^{m} g_i(x(t))u^*_{s,i}(0|t_k) + k(x(t))w(t)\right)\\ &-\frac{\partial V}{\partial x}(f(x(t_k)) + \sum_{i=1}^{m} g_i(x(t_k))u^*_{s,i}(0|t_k))\end{aligned} \tag{13}$$

Taking into account Eqs. 2 and 3, the following inequality if obtained for all $x(t_k) \in \Omega_\rho/\Omega_{\rho_s}$[§] from Eq. 13

$$\begin{aligned}\dot{V}(x(t)) \leq& -\alpha_3(\alpha_2^{-1}(\rho_s))\\ &+\left(L_x + \sum_{i=1}^{m} L_{u_i}u^*_{s,i}(0|t_k)\right)|x(t) - x(t_k)|\\ &+L_w|w(t)|.\end{aligned} \tag{14}$$

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

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

Using this expression, the bounds on the disturbance $w(t)$ and the inputs $u_i$, $i = 1,\ldots, m$, and Eq. 14, 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 $x(t_k) \in \Omega_\rho/\Omega_{\rho_s}$

§The operator "/" is used to denote set subtraction, i.e., $A/B := \{x \in R^{n_x}: x \in A, x \notin B\}$.

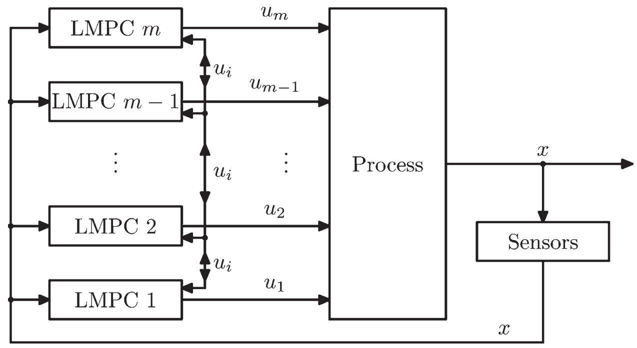


**Figure 3. Iterative distributed LMPC.**

$$\dot{V}(x(t)) \leq -\alpha_3(\alpha_2^{-1}(\rho_s)) + \left(L_x + \sum_{i=1}^{m} L_{u_i}u_i^{\max}\right)M + L_w\theta. \tag{15}$$

If the condition of Eq. 9 is satisfied, then there exists $\varepsilon_w > 0$ such that the following inequality holds for $x(t_k) \in \Omega_\rho/\Omega_{\rho s}$

$$\dot{V}(x(t)) \leq -\varepsilon_w/\Delta \tag{16}$$

for $t \in [t_k, t_{k+1})$ . Integrating the inequality of Eq. 16 on $t \in [t_k, t_{k+1})$, we obtain that

$$V(x(t_{k+1}) \leq V(x(t_k)) - \varepsilon_w \tag{17}$$

and

$$V(x(t)) \leq V(x(t_k)),\ \forall t \in [t_k, t_{k+1}) \tag{18}$$

for all $x(t_k) \in \Omega_\rho/\Omega_{\rho_s}$. Using Eqs. 17 and 18 recursively it can be proved that, if $x(t_0) \in \Omega_\rho/\Omega_{\rho_s}$, the state converges to $\Omega_{\rho_s}$ 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 $\Omega_{\rho^*}$ for all times. This statement holds because of the definition of $\rho^*$. This proves that the closed-loop system under the proposed distributed LMPC of Eqs. 7 and 8 is ultimately bounded in $\Omega_{\rho^*}$. ■

### *Iterative distributed LMPC*

An alternative to the sequential distributed LMPC architecture presented in the previous subsection is to evaluate all the distributed LMPCs in parallel and iterate to improve closed-loop performance. A schematic of this control architecture is shown in Figure 3. In this architecture, each distributed LMPC controller must be able to communicate with all the other controllers (i.e., the distributed controllers are all interconnected). More specifically, when a new state measurement is available at a sampling time, each distributed LMPC controller evaluates and obtains its future input trajectory; and then each LMPC controller broadcasts its latest obtained future input trajectory to all the other controllers. On the basis of the newly received input trajectories, each LMPC controller evaluates its future input trajectory again and this process is repeated until a certain termination condition is satisfied. Specifically, we proposed to use the following implementation strategy:

1. At each sampling time $t_k$, all the LMPC controllers receive the state measurement $x(t_k)$ from the sensors.
2. At iteration $c$ ($c \geq 1$):
   2.1. All the distributed LMPC controllers exchange their latest future input trajectories.
   2.2. Each LMPC controller evaluates its own future input trajectory based on $x(t_k)$ and the latest received input trajectories of all the other LMPC controllers.
3. If the termination condition is satisfied, each LMPC controller sends the first step input value of its latest input trajectory to its actuators; if the termination condition is not satisfied, go to step 2 ($c = c + 1$).

Note that at the initial iteration, all the LMPC controllers 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 proposed distributed LMPC architecture; a point that will be made clear below. For the iterations in this distributed LMPC 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., $c \leq c_{\max}$) 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 or the iterations maybe terminated when a maximum computational time is reached.

To proceed, we define $\hat{x}(\tau|t_k)$ for $\tau \in [0, N\Delta)$ as the nominal sampled trajectory of the system of Eq. 1 associated with the feedback control law $h(x)$ and sampling time $\Delta$ starting from $x(t_k)$. This nominal sampled trajectory is obtained by integrating recursively the following differential equation:

$$\dot{\hat{x}}(\tau|t_k) = f(\hat{x}(\tau|t_k)) + \sum_{i=1}^{m} g_i(\hat{x}(\tau|t_k))h_i(\hat{x}(l\Delta|t_k)),$$
$$\forall \tau \in ((l\Delta, (l+1)\Delta)),\ l = 0, \ldots, N-1.$$

On the basis of $\hat{x}(\tau|t_k)$, we can define the following variable

$$u_{p,j}^{*,0}(\tau|t_k) = h_j(\hat{x}(l\Delta|t_k)),$$
$$j = 1, \ldots, m,\ \forall \tau \in ((l\Delta, (l+1)\Delta)),\ l = 0, \ldots, N-1,$$

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

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

$$u_{p,j}^{*,c}(\tau|t_k) = \arg \min_{u_{p,j} \in S(\Delta)} \int_0^{N\Delta} (\tilde{x}^T(\tau)Q_c\tilde{x}(\tau) + \sum_{i=1}^{m} u_{p,i}(\tau)^T R_{ci} u_{p,i}(\tau)]d\tau \quad (19a)$$

$$\text{s.t. } \dot{\tilde{x}}(\tau) = f(\tilde{x}(\tau)) + \sum_{i=1}^{m} g_i(\tilde{x}(\tau))u_{p,i} \quad (19b)$$

$$u_{p,i}(\tau) = u_{p,i}^{*,c-1}(\tau|t_k),\ \forall i \neq j \quad (19c)$$

$$u_{p,j}(\tau) \in U_j \quad (19d)$$

$$\tilde{x}(0) = x(t_k) \quad (19e)$$

$$\frac{\partial V(x)}{\partial x} g_j(x(t_k))u_{p,j}(0) \leq \frac{\partial V(x)}{\partial x} g_j(x(t_k))h_j(x(t_k)) \quad (19f)$$

where $\tilde{x}$ is the predicted trajectory of the nominal system with $u_k$, the input trajectory, computed by the LMPCs of Eq. 19 and all the other inputs are the optimal input trajectories at iteration $c - 1$ of the rest of distributed controllers. The optimal solution to the optimization problem of Eq. 19 is denoted $u_{p,j}^{*,c}(\tau|t_k)$, which is defined for $\tau \in [0,N\Delta)$. Accordingly, we define the final optimal input trajectory of LMPC $j$ (that is, the optimal trajectories computed at the last iteration) as $u_{p,j}^{*}(\tau|t_k)$, which is also defined for $\tau \in [0, N\Delta)$.

The manipulated inputs of the proposed control design of Eq. 19 are defined as follows:

$$u_i(t) = u_{p,i}^{*}(t - t_k|t_k),\ i = 1, \ldots, m,\ \forall t \in [t_k, t_{k+1}). \quad (20)$$

In what follows, we refer to this distributed LMPC architecture as the iterative distributed LMPC. The stability property of the iterative distributed LMPC is stated in the following Theorem 2.

**Theorem 2.** *Consider the system of* Eq. 1 *in closed-loop under the distributed LMPC of Eqs. 19 and 20 based on the controller $h(x)$ that satisfies the condition of* Eq. 2. *Let $\varepsilon_w > 0$, $\Delta > 0$ and $\rho > \rho_s > 0$ satisfy the constraint of Eq. 9. For any $N \geq 1$ and $c \geq 1$, if $x(t_0) \in \Omega_\rho$ and if $\rho^* \leq \rho$ where $\rho^*$ is defined as in Eq.* 10, *then the state $x(t)$ of the closed-loop system is ultimately bounded in $\Omega_{\rho^*}$.*

**Proof.** Similar to the proof of Theorem 1, the proof of Theorem 2 also consists of two parts. We first prove that the optimization problem of Eq. 19 is feasible for each iteration $c$ and $x \in \Omega_\rho$. Then we prove that, under the proposed distributed LMPC scheme of Eqs. 19 and 20, the state of the system of Eq. 1 is ultimately bounded in $\Omega_{\rho^*}$.

**Part 1:** To prove the feasibility of the optimization problem of Eq. 19, we only have to prove that there exists a $u_{p,j}(0)$ which satisfies the input constraint of Eq. 19d and the constraint of Eq. 19f. This is because the constraint of Eq. 19f is only enforced on the first prediction step of $u_{p,j}(\tau)$ and in the prediction time $\tau \in [\Delta, N\Delta)$, the input constraint of Eq. 20 can be easily satisfied with $u_{p,j}(\tau)$ being any value in the convex set $U_j$.

We assume that $x(t_k) \in \Omega_\rho$ ($x(t)$ is bounded in $\Omega_\rho$, which will be proved in Part 2). It is easy to verify that the value of $u_{p,j}$ such that $u_{p,j}(0) = h_j(x(t_k))$ satisfies the input constraint of Eq. 19d (assumed property of $h(x)$ for $x \in \Omega_\rho$) and the constraint of Eq. 19f for all possible $c$, thus, the feasibility of LMPC $j$, $j = 1,\ldots, m$, is guaranteed.

**Part 2.** By adding the constraints of Eq. 19f of each LMPC together, we have

$$\sum_{j=1}^{m} \frac{\partial V(x)}{\partial x} g_j(x(t_k))u_{p,j}^{*,c}(0|t_k) \leq \sum_{j=1}^{m} \frac{\partial V(x)}{\partial x} g_j(x(t_k))h_j(x(t_k))$$

It follows from the above inequality and condition of Eq. 2 that

$$\frac{\partial V}{\partial x}\left(f(x(t_k)) + \sum_{j=1}^{m} g_j(x(t_k))u^{*,c}_{p,j}(0|t_k)\right)$$
$$\le \frac{\partial V}{\partial x}\left(f(x(t_k)) + \sum_{j=1}^{m} g_j(x(t_k))h_j(x(t_k))\right)$$
$$\le -\alpha_3(|x(t_k)|). \quad (21)$$

Following the same approach as in the proof of Theorem 1, we know that if condition of Eq. 9 is satisfied, then the state of the closed-loop system can be proved to be maintained in $\Omega_{\rho^*}$ under the proposed distributed MPC architecture of Eqs. 19 and 20. ■

**Remark 5.** Note that the distributed MPC designs have the same stability region $\Omega_\rho$ as the one of the Lyapunov-based controller $h(x)$. When the stability of the Lyapunov-based controller $h(x)$ is global (i.e., the stability region is the entire state space), then the stability of the distributed MPC designs is also global. Note also that for any initial condition in $\Omega_\rho$, the distributed MPC designs are proved to be feasible.

**Remark 6.** We do not consider delays introduced into the system by the communication network or by the time needed to solve the optimization problems. In future studies, these delays will be taken into account in the formulation of the controllers. In this study, state constraints have also not been considered but the proposed designs can be extended to handle state constraints by restricting the closed-loop stability region further to satisfy the state constraints.

**Remark 7.** The choice of the horizon of the distributed MPC designs does not affect the stability of the closed-loop system. For any horizon length $N \ge 1$, the closed-loop stability is guaranteed by the constraints of Eqs. 7g and 19f. However, the choice of the horizon does affect the performance of the distributed MPC designs.

**Remark 8.** Note that because the manipulated inputs enter the dynamics of the system of Eq. 1 in an affine manner, the constraints designed in the LMPC optimization problems of Eqs. 7 and 19 to guarantee the closed-loop stability can be decoupled for different distributed controllers as in Eqs. 7g and 19f.

**Remark 9.** In the sequential distributed LMPC architecture the distributed controllers are evaluated in sequence, which implies that the minimal time to obtain a set of solutions to all the LMPC controllers is the sum of the evaluation times of all the LMPC controllers; whereas in the iterative distributed LMPC architecture the distributed controllers are evaluated in parallel, which implies that the minimal time to obtain a set of solutions to all the LMPC controllers in each iteration is the largest evaluation time among all the LMPCs.

**Remark 10.** Note that the sequential (or iterative) distributed LMPC is not a direct decomposition of the centralized LMPC because the set of constraints of Eq. 7g (or Eq. 19f) for $j = 1,\ldots, m$ in the distributed LMPC formulation of Eq. 7 (or Eq. 19) imposes a different feasibility region from the one of the centralized LMPC of Eq. 5, which has a single constraint (Eq. 5e).

**Remark 11.** In general, there is no guaranteed convergence of the optimal cost or solution of an iterated distributed MPC 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. The reader may refer to References 3 and 7 for discussions on the conditions under which convergence of the solution of a distributed linear or convex MPC design to the solution of a centralized MPC or a Pareto optimal solution is ensured in the context of linear systems.

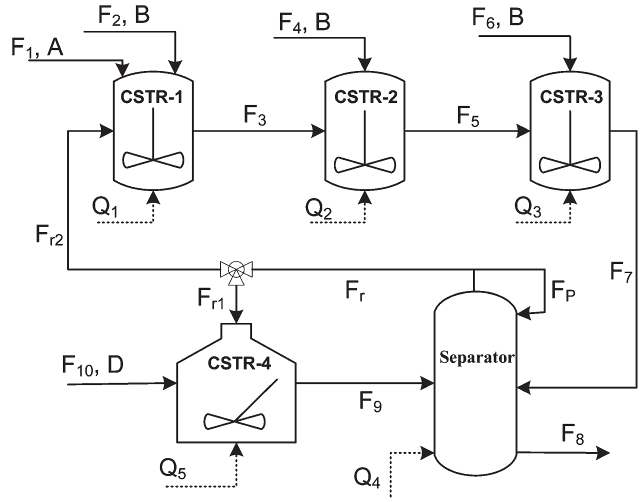


**Figure 4. Process flow diagram of alkylation of benzene.**

**Remark 12.** Note also that, in general, there is no guarantee that the closed-loop performance of one (centralized or distributed) MPC architecture discussed in this study should be superior than the others since the solutions provided by these MPC architectures are proved to be feasible and stabilizing but the superiority of the performance of one MPC architecture over another is not established. This is because the MPC designs are implemented in a receding horizon scheme and the prediction horizon is finite; and also because the different MPC designs are not equivalent as we discussed in Remark 10 and the non-convexity property as we discussed in Remark 11. In applications of these MPC architectures, especially for chemical process control in which non-convex problems is a very common occurrence, simulations should be conducted before making decisions as to which architecture should be used.

## Application to a Chemical Process Example

The process of alkylation of benzene with ethylene to produce ethylbenzene is widely used in the petrochemical industry. Dehydration of the product produces styrene, which is the precursor to polystyrene and many copolymers. Over the last 2 decades, several methods and simulation results of alkylation of benzene with catalysts have been reported in the literature. The process model developed in this section is based on the References 26–29. More specifically, the process considered in this study consists of four continuously stirred tank reactors (CSTRs) and a flash tank separator, as shown in Figure 4. The CSTR-1, CSTR-2, and CSTR-3 are in series and involve the alkylation of benzene with ethylene. Pure benzene is fed from stream $F_1$ and pure ethylene is fed from 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 further distillation process that we do not consider in this example. In CSTR-4, reaction 2 and catalyzed transalkylation reaction in which 1,3-diethylbenzene reacts with benzene to produce ethylbenzene (reaction 3) takes 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 dynamic equations describing the behavior of the process, obtained through material and energy balances under standard modeling assumptions, are given below:

$$\frac{dC_{A1}}{dt} = \frac{F_1 C_{A0} + F_{r2} C_{Ar} - F_3 C_{A1}}{V_1} - r_1(T_1, C_{A1}, C_{B1}) \quad (22a)$$

$$\frac{dC_{B1}}{dt} = \frac{F_2 C_{B0} + F_{r2} C_{Br} - F_3 C_{B1}}{V_1} - r_1(T_1, C_{A1}, C_{B1}) - r_2(T_1, C_{B1}, C_{C1}) \quad (22b)$$

$$\frac{dC_{C1}}{dt} = \frac{F_{r2} C_{Cr} - F_3 C_{C1}}{V_1} + r_1(T_1, C_{A1}, C_{B1}) - r_2(T_1, C_{B1}, C_{C1}) \quad (22c)$$

$$\frac{dC_{D1}}{dt} = \frac{F_{r2} C_{Dr} - F_3 C_{D1}}{V_1} + r_2(T_1, C_{B1}, C_{C1}) \quad (22d)$$

$$\frac{dT_1}{dt} = \frac{Q_1 + F_1 C_{A0} H_A(T_{A0}) + F_2 C_{B0} H_B(T_{B0}) + \sum_i^{A,B,C,D} \left(F_{r2} C_{ir} H_i(T_4) - F_3 C_{i1} H_i(T_1)\right)}{\sum_i^{A,B,C,D} C_{i1} C_{pi} V_1} + \frac{-\Delta H_{r1} r_1(T_1, C_{A1}, C_{B1}) - \Delta H_{r2} r_2(T_1, C_{B1}, C_{C1})}{\sum_i^{A,B,C,D} C_{i1} C_{pi} V_1} \quad (22e)$$

$$\frac{dC_{A2}}{dt} = \frac{F_3 C_{A1} - F_5 C_{A2}}{V_2} - r_1(T_2, C_{A2}, C_{B2}) \quad (22f)$$

$$\frac{dC_{B2}}{dt} = \frac{F_3 C_{B1} + F_4 C_{B0} - F_5 C_{B2}}{V_2} - r_1(T_2, C_{A2}, C_{B2}) - r_2(T_2, C_{B2}, C_{C2}) \quad (22g)$$

$$\frac{dC_{C2}}{dt} = \frac{F_3 C_{C1} - F_5 C_{C2}}{V_2} + r_1(T_2, C_{A2}, C_{B2}) - r_2(T_2, C_{B2}, C_{C2}) \quad (22h)$$

$$\frac{dC_{D2}}{dt} = \frac{F_3 C_{D1} - F_5 C_{R2}}{V_2} + r_2(T_2, C_{B2}, C_{C2}) \quad (22i)$$

$$\frac{dT_2}{dt} = \frac{Q_2 + F_4 C_{B0} H_B(T_{B0}) + \sum_i^{A,B,C,D} \left(F_3 C_{i1} H_i(T_1) - F_5 C_{i2} H_i(T_2)\right)}{\sum_i^{A,B,C,D} C_{i2} C_{pi} V_2} + \frac{-\Delta H_{r1} r_1(T_2, C_{A2}, C_{B2}) - \Delta H_{r2} r_2(T_2, C_{A2}, C_{B2})}{\sum_i^{A,B,C,D} C_{i2} C_{pi} V_2} \quad (22j)$$

$$\frac{dC_{A3}}{dt} = \frac{F_5 C_{A2} - F_7 C_{A3}}{V_3} - r_1(T_3, C_{A3}, C_{B3}) \quad (22k)$$

$$\frac{dC_{B3}}{dt} = \frac{F_5 C_{B2} + F_6 C_{B0} - F_7 C_{B3}}{V_3} - r_1(T_3, C_{A3}, C_{B3}) - r_2(T_3, C_{B3}, C_{C3}) \quad (22l)$$

$$\frac{dC_{C3}}{dt} = \frac{F_5 C_{C2} - F_7 C_{C3}}{V_3} + r_1(T_3, C_{A3}, C_{B3}) - r_2(T_3, C_{B3}, C_{C3}) \quad (22m)$$

$$\frac{dC_{D3}}{dt} = \frac{F_5 C_{D2} - F_7 C_{D3}}{V_3} + r_2(T_3, C_{B3}, C_{C3}) \quad (22n)$$

$$\frac{dT_3}{dt} = \frac{Q_3 + F_6 C_{B0} H_B(T_{B0}) + \sum_i^{A,B,C,D} \left(F_5 C_{i2} H_i(T_2) - F_7 C_{i3} H_i(T_3)\right)}{\sum_i^{A,B,C,D} C_{i3} C_{pi} V_3} + \frac{-\Delta H_{r1} r_1(T_3, C_{A3}, C_{B3}) - \Delta H_{r2} r_2(T_3, C_{B3}, C_{C3})}{\sum_i^{A,B,C,D} C_{i3} C_{pi} V_3} \quad (22o)$$

$$\frac{dC_{A4}}{dt} = \frac{F_7 C_{A3} + F_9 C_{A5} - F_r C_{Ar} - F_8 C_{A4}}{V_4} \quad (22p)$$

$$\frac{dC_{B4}}{dt} = \frac{F_7 C_{B3} + F_9 C_{B5} - F_r C_{Br} - F_8 C_{B4}}{V_4} \quad (22q)$$

$$\frac{dC_{C4}}{dt} = \frac{F_7 C_{C3} + F_9 C_{C5} - F_r C_{Cr} - F_8 C_{C4}}{V_4} \quad (22r)$$

$$\frac{dC_{D4}}{dt} = \frac{F_7 C_{D3} + F_9 C_{D5} - F_r C_{Dr} - F_8 C_{D4}}{V_4} \quad (22s)$$

$$\frac{dT_4}{dt} = \frac{Q_4 + \sum_i^{A,B,C,D} \left(F_7 C_{i3} H_i(T_3) + F_9 C_{i5} H_i(T_5) - F_r C_{ir} H_i(T_4) - F_8 C_{i4} H_i(T_4) - F_r C_{ir} H_{\text{vapi}}\right.}{\sum_i^{A,B,C,D} C_{i4} C_{pi} V_4} \quad (22t)$$

**Table 1. Process Variables**

| | |
|---|---|
| $C_{A1}$, $C_{B1}$, $C_{C1}$, $C_{D1}$ | Concentrations of $A$, $B$, $C$, $D$ in CSTR-1 |
| $C_{A2}$, $C_{B2}$, $C_{C2}$, $C_{D2}$ | Concentrations of $A$, $B$, $C$, $D$ in CSTR-2 |
| $C_{A3}$, $C_{B3}$, $C_{C3}$, $C_{D3}$ | Concentrations of $A$, $B$, $C$, $D$ in CSTR-3 |
| $C_{A4}$, $C_{B4}$, $C_{C4}$, $C_{D4}$ | Concentrations of $A$, $B$, $C$, $D$ in separator |
| $C_{A5}$, $C_{B5}$, $C_{C5}$, $C_{D5}$ | Concentrations of $A$, $B$, $C$, $D$ in CSTR-4 |
| $C_{Ar}$, $C_{Br}$, $C_{Cr}$, $C_{Dr}$ | Concentrations of $A$, $B$, $C$, $D$ in $F_r$, $F_{r1}$, $F_{r2}$ |
| $T_1$, $T_2$, $T_3$, $T_4$, $T_5$ | Temperatures in each vessel |
| $T_{ref}$ | Reference temperature |
| $F_3$, $F_5$, $F_7$, $F_8$, $F_9$ | Effluent flow rates from each vessel |
| $F_1$, $F_2$, $F_4$, $F_6$, $F_10$ | Feed flow rates to each vessel |
| $F_r$, $F_{r1}$, $F_{r2}$ | Recycle flow rates |
| $H_{vapA}$, $H_{vapB}$, $H_{vapC}$, $H_{vapD}$ | Enthalpies of vaporization of $A$, $B$, $C$, $D$ |
| $H_{Aref}$, $H_{Bref}$, $H_{Cref}$, $H_{Dref}$ | Enthalpies of $A$, $B$, $C$, $D$ at $T_{ref}$ |
| $\Delta H_{r1}$, $\Delta H_{r2}$, $\Delta H_{r3}$ | Heat of reactions 1, 2, and 3 |
| $V_1$, $V_2$, $V_3$, $V_4$, $V_5$ | Volume of each vessel |
| $Q_1$, $Q_2$, $Q_3$, $Q_4$, $Q_5$ | External heat/coolant inputs to each vessel |
| $C_{pA}$, $C_{pB}$, $C_{pC}$, $C_{pD}$ | Heat capacity of $A$, $B$, $C$, $D$ at liquid phase |
| $\alpha_A$, $\alpha_B$, $\alpha_C$, $\alpha_D$ | Relative volatilities of $A$, $B$, $C$, $D$ |
| $C_{A0}$, $C_{B0}$, $C_{C0}$, $C_{D0}$ | Molar densities of pure $A$, $B$, $C$, $D$ |
| $T_{A0}$, $T_{B0}$, $T_{D0}$ | Feed temperatures of pure $A$, $B$, $D$ |

$$\frac{dC_{A5}}{dt} = \frac{F_{r1}C_{Ar} - F_9 C_{A5}}{V_5} - r_3(T_5, C_{A5}, C_{D5}) \quad (22u)$$

$$\frac{dC_{B5}}{dt} = \frac{F_{r1}C_{Br} - F_9 C_{B5}}{V_5} - r_2(T_5, C_{B5}, C_{C5}) \quad (22v)$$

$$\frac{dC_{C5}}{dt} = \frac{F_{r1}C_{Cr} - F_9 C_{C5}}{V_5} - r_2(T_5, C_{B5}, C_{C5}) + 2r_3(T_5, C_{A5}, C_{D5}) \quad (22w)$$

$$\frac{dC_{D5}}{dt} = \frac{F_{r1}C_{Dr} + F_{10}C_{D0} - F_9 C_{D5}}{V_5} + r_2(T_5, C_{B5}, C_{C5}) - r_3(T_5, C_{A5}, C_{D5}) \quad (22x)$$

$$\frac{dT_5}{dt} = \frac{Q_5 + F_{10}C_{D0}H_D(T_{D0}) + \sum_i^{A,B,C,D}(F_r C_{ir} H_i)(T_4) - F_9 C_{i5} H_i(T_5)}{\sum_i^{A,B,C,D} C_{i5} C_{pi} V_5} + \frac{-\Delta H_{r2} r_2(T_5, C_{B5}, C_{C5}) - \Delta H_{r3} r_3(T_5, C_{A5}, C_{D5})}{\sum_i^{A,B,C,D} C_{i5} C_{pi} V_5} \quad (22y)$$

where $r_1$, $r_2$, and $r_3$ are the reaction rates of reactions 1, 2, and 3, respectively, and $H_i$, $i = A, B, C, D$, are the enthalpies of the reactants. The reaction rates are related to the concentrations of the reactants and the temperature in each reactor as follows:

$$r_1(T, C_A, C_B) = k_{r1} C_A^{0.32} C_B^{1.5}, \quad r_2(T, C_B, C_C) = \frac{k_{r2} C_B^{2.5} C_C^{0.5}}{(1 + k_{EB2} C_D)},$$
$$r_3(T, C_A, C_D) = \frac{k_{r3} C_A^{1.0218} C_D}{(1 + k_{EB3} C_A)}$$

with

$$k_{r1} = 0.0840 e^{(-9502/RT)}, \; k_{r2} = 0.0850 e^{(-20640/RT)},$$
$$k_{r3} = 237.8 e^{(-61280/RT)}, \; k_{EB2} = 0.0152 e^{(-3933/RT)},$$
$$k_{EB3} = 0.4901 e^{(-50870/RT)}$$

**Table 2. Parameter Values**

| | | | |
|---|---|---|---|
| $F_1 = 7.1 \times 10^{-3}$ | $m^3/s$ | $F_r = 0.012$ | $m^3/s$ |
| $F_2 = 8.697 \times 10^{-4}$ | $m^3/s$ | $F_{r1} = 0.006$ | $m^3/s$ |
| $F_{r2} = 0.006$ | $m^3/s$ | $V_1 = 1$ | $m^3$ |
| $F_10 = 2.31 \times 10^{-3}$ | $m^3/s$ | $V_2 = 1$ | $m^3$ |
| $H_{vapA} = 3.073 \times 10^4$ | J/mol | $V_3 = 1$ | $m^3$ |
| $H_{vapB} = 1.35 \times 10^4$ | J/mol | $V_4 = 3$ | $m^3$ |
| $H_{vapC} = 4.226 \times 10^4$ | J/mol | $V_5 = 1$ | $m^3$ |
| $H_{vapD} = 4.55 \times 10^4$ | J/mol | $C_{pA} = 184.6$ | J/mol K |
| $\Delta H_{r1} = -1.536 \times 10^5$ | J/mol | $C_{pB} = 59.1$ | J/mol K |
| $\Delta H_{r2} = -1.118 \times 10^5$ | J/mol | $C_{pC} = 247$ | J/mol K |
| $\Delta H_{r3} = 4.141 \times 10^5$ | J/mol | $C_{pD} = 301.3$ | J/mol K |
| $C_{A0} = 1.126 \times 10^4$ | $Mol/m^3$ | $T_{ref} = 450$ | K |
| $C_{B0} = 2.028 \times 10^4$ | $Mol/m^3$ | $T_{A0} = 473$ | K |
| $C_{C0} = 8174$ | $Mol/m^3$ | $T_{B0} = 473$ | K |
| $C_{D0} = 6485$ | $Mol/m^3$ | $T_{D0} = 473$ | K |

The heat capacities of the species are assumed to be constants and the molar enthalpies have a linear dependence on temperature as follows:

$$H_i(T) = H_{iref} + C_{pi}(T - T_{ref}), \quad i = A, B, C, D$$

where $C_{pi}$ , $i = A, B, C, D$ are heat capacities.

The model of the flash tank separator is developed under the assumption that the relative volatility of each species has a linear correlation with the temperature of the vessel within the operating temperature range of the flash tank, as shown below:

$$\alpha_A = 0.0449T_4 + 10, \quad \alpha_B = 0.0260T_4 + 10$$
$$\alpha_C = 0.0065T_4 + 0.5, \quad \alpha_D = 0.0058T_4 + 0.25$$

where $\alpha_i$, $i = A, B, C, D$, represent the relative volatilities. 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:

$$M_i = (F_7 C_{i3} + F_9 C_{i5}) \frac{\alpha_i (F_7 C_{i3} + F_9 C_{i5})}{\sum_k^{A,B,C,D} \alpha_k (F_7 C_{k3} + F_9 C_{k5})},$$
$$i = A, B, C, D$$

where $M_i$, $i = A, B, C, D$ are the molar flow rates of the overhead reactants. On the basis of $M_i$, $i = A, B, C, D$, we can calculate the concentration of the reactants in the recycle streams as follows:

$$C_{ir} = \frac{M_i}{\sum_k^{A,B,C,D} M_i / C_{k0}}, \quad i = A, B, C, D$$

where $C_{k0}$, $k = A, B, C, D$, are the mole densities of pure reactants. The condensation of vapor takes place overhead, and a portion of the condensed liquid is purged back to separator to keep the flow rate of the recycle stream at a fixed value. The temperature of the condensed liquid is assumed to be the same as the temperature of the vessel.

**Table 3. Steady-State Input Values for $x_s$**

| | | | | | |
|---|---|---|---|---|---|
| $Q_{1s}$ | $-4.4 \times 10^6$ [J/s] | $Q_{2s}$ | $-4.6 \times 10^6$ [J/s] | $Q_{3s}$ | $-4.7 \times 10^6$ [J/s] |
| $Q_{4s}$ | $9.2 \times 10^6$ [J/s] | $Q_{5s}$ | $5.9 \times 10^6$ [J/s] | $F_{4s}$, $F_{6s}$ | $8.697 \times 10^{-4}$ [$m^3/s$] |

**Table 4. Manipulated Input Constraints**

| | | |
|---|---|---|
| $\lvert u_{11}\rvert \le 7.5 \times 10^5$ [J/s] | $\lvert u_{1i}\rvert \le 5 \times 10^5$ [J/s],($i$ = 2,3) | $\lvert u_{21}\rvert \le 6 \times 10^5$ [J/s] |
| $\lvert u_{22}\rvert \le 5 \times 10^5$ [J/s] | $\lvert u_{31}\rvert \le 4.93 \times 10^{-5}$ [m$^3$/s] | $\lvert u_{32}\rvert \le 4.93 \times 10^{-5}$ [m$^3$/s] |

The definitions for the variables used in the aforementioned model can be found in Table 1, with the parameter values given in Table 2.

Each of the tanks has an external heat/coolant input. The manipulated inputs to the process are the heat injected to or removed from the five vessels, $Q_1$, $Q_2$, $Q_3$, $Q_4$, and $Q_5$, and the feed stream flow rates to CSTR-2 and CSTR-3, $F_4$ and $F_6$.

The states of the process consist of the concentrations of $A$, $B$, $C$, $D$ in each of the five vessels and the temperatures of the vessels. The state of the process is assumed to be available continuously to the controllers. We consider a stable steady state (operating point), $x_s$, of the process which is defined by the steady-state inputs $Q_{1s}$, $Q_{2s}$, $Q_{3s}$, $Q_{4s}$, $Q_{5s}$, $F_{4s}$, and $F_{6s}$, which are shown in Table 3. The steady-state temperatures in the five vessels are the following:

$$T_{1s} = 477.24\text{K},\ T_{2s} = 476.97\text{K},\ T_{3s} = 473.47\text{K},\ T_{4s} = 470.60\text{K},\ T_{5s} = 478.28\text{K}.$$

The control objective is to regulate the system from an initial state to the steady state. The temperatures of the five vessels in the initial state are as follows:

$$T_{1o} = 443.02\,K,\ T_{2o} = 437.12\,\text{K},\ T_{3o} = 428.37\,\text{K},\ T_{4o} = 433.15\,\text{K},\ T_{5o} = 457.55\,\text{K}.$$

The first distributed controller (LMPC 1) will be designed to decide the values of $Q_1$, $Q_2$, and $Q_3$, the second distributed controller (LMPC 2) will be designed to decide the values of $Q_4$ and $Q_5$, and the third distributed controller (LMPC 3) will be designed to decide the values of $F_4$ and $F_6$. Taking this into account, the process model of Eq. 22a belongs to the following class of nonlinear systems:

$$\dot{x}(t) = f(x) + g_1(x)u_1(x) + g_2(x)u_2(x) + g_3(x)u_3(x)$$

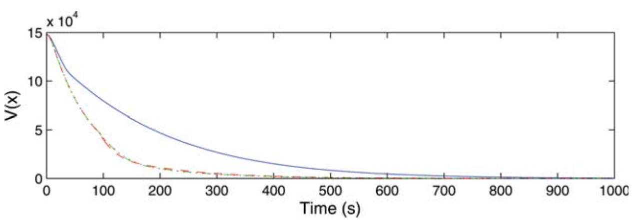


**Figure 5. Trajectories of the Lyapunov function *V*(*x*) under the Lyapunov-based controller *h*(*x*) implemented in a sample-and-hold fashion (solid line), the centralized LMPC (dashed line), the sequential distributed LMPC (dash-dotted line), and the iterative distributed LMPC with *c* = 1 (dotted line).**

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

**Table 5. Mean Evaluation Time of Different LMPC Optimization Problems for 100 Evaluations**

| | | $N = 1$ (s) | $N = 3$ (s) | $N = 6$ (s) |
|---|---|---|---|---|
| Centralized LMPC | | 2.192 | 8.694 | 27.890 |
| Sequential | LMPC 1 | 0.472 | 2.358 | 6.515 |
| | LMPC 2 | 0.497 | 1.700 | 4.493 |
| | LMPC 3 | 0.365 | 1.453 | 3.991 |
| Iterative | LMPC 1 | 0.484 | 2.371 | 6.280 |
| | LMPC 2 | 0.426 | 1.716 | 4.413 |
| | LMPC 3 | 0.185 | 0.854 | 2.355 |

where the state $x$ is the deviation of the state of the process from the steady state, $u_1^T = [u_{11}\ u_{12}\ u_{13}] = [Q_1 - Q_{1s}\ Q_2 - Q_{2s}\ Q_3 - Q_{3s}]$, $u_2^T = [u_{21}\ u_{22}] = [Q_4 - Q_{4s}\ Q_5 - Q_{5s}]$ and $u_3^T = [u_{31}\ u_{32}] = [F_4 - F_{4s}\ F_6 - F_{6s}]$ are the manipulated inputs which are subject to the constraints shown in Table 4.

In the control of the process, $u_1$ and $u_2$ are necessary to keep the stability of the closed-loop system, whereas $u_3$ can be used as an extra manipulated input to improve the closed-loop performance. To illustrate the theoretical results, we first design the Lyapunov-based controller $h(x) = [h_1(x)\ h_2(x)\ h_3(x)]^T$. Specifically, $h_1(x)$ and $h_2(x)$ are designed as follows[30]:

$$h_i(x) = \begin{cases} -\frac{L_fV + \sqrt{(L_fV)^2 + (L_{g_i}V)^4}}{(L_{g_i}V)^2} L_{g_i}V & \text{if} \quad L_{g_i}V \neq 0 \\ 0 & \text{if} \quad L_{g_i}V = 0 \end{cases}$$

where $i$ = 1, 2, $L_fV = \frac{\partial V}{\partial x} f(x)$ and $L_{g_i}V = \frac{\partial V}{\partial x} g_i(x)$ denote the Lie derivatives of the scalar function $V$ with respect to the vector fields $f$ and $g_i$ ($i$ = 1, 2), respectively. The controller $h_3(x)$ is chosen to be $h_3(x) = [0\ 0]^T$ because the input set $u_3$ is not needed to stabilize the process. We consider a Lyapunov function $V(x) = x^T Px$ with $P$ being the following weight matrix[¶]

$$P = \text{diag}^{\P}([1\,1\,1\,1\,10\,1\,1\,1\,1\,10\,1\,1\,1\,1\,10\,1\,1\,1\,1\,10\,1\,1\,1\,1\,10]).$$

The weights in $P$ are chosen by a trail-and-error procedure. The basic idea behind this procedure is that more weight should be put on the temperatures of the five vessels because temperatures have more significant effect on the overall control performance, and the Lyapunov-based controller $h(x)$ should be able to stabilize the closed-loop system asymptotically with continuous feedback and actuation.

On the basis of $h(x)$, we design the centralized LMPC, the sequential distributed LMPC, and the iterative distributed LMPC. The sampling time used is $\Delta$ = 30 s and the weight matrices

$$Q_c = \text{diag}([1\,1\,1\,1\,10^3\,1\,1\,1\,1\,10^3\,10\,10\,10\,10\,10^4\,1\,1\,1\,1\,10^3\,1\,1\,1\,1\,10^3]).$$

and $R_{c1} = \text{diag}([10^{-8}\ 10^{-8}\ 10^{-8}])$, $R_{c2} = \text{diag}([10^{-8}\ 10^{-8}])$, and $R_{c3} = \text{diag}([1\ 1])$.

First, we carried out a set of simulations which demonstrate that the Lyapunov-based controller and the different schemes of LMPCs can all stabilize the closed-loop system asymptotically. Figure 5 shows the trajectories of the Lyapu-

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

**Table 6. Total Performance Cost Along the Closed-Loop System Trajectories I**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | J ($\times 10^7$) | | | | | | | |
| Centralized | 1.8858 | | | | | | | |
| Sequential | 1.8891 | | | | | | | |
| $c_{max}$ | 1 | 3 | 5 | 7 | 9 | 11 | 13 | 15 |
| Iterative | 1.8955 | 1.8883 | 1.8867 | 1.8863 | 1.8862 | 1.8859 | 1.8858 | 1.8858 |

nov function $V(x)$ under the different control schemes. Note that because of the constraints of Eqs. 5e, 7g, and 19f, the trajectories of the Lyapunov function of the closed-loop system under the centralized LMPC, the sequential distributed LMPC and the iterative distributed LMPC are guaranteed to be bounded by the corresponding Lyapunov function trajectory under the Lyapunov-based controller $h(x)$ implemented in a sample-and-hold fashion with the sampling time $\Delta$. This point is also illustrated in Figure 5.

Next, we compare the mean evaluation times of the centralized LMPC optimization problem and the sequential and iterative distributed LMPC optimization problems. Each LMPC optimization problem was evaluated 100 times at different conditions. Different prediction horizons were considered in this set of simulations. The simulations were carried out using Java programming language in a Pentium 3.20 GHz computer. The optimization problems were solved using the open source interior point optimizer Ipopt.[31] The results are shown in Table 5. From Table 5, we can see that in all cases, the time needed to solve the centralized LMPC is much larger than the time needed to solve the sequential or iterative distributed LMPCs. This is because the centralized LMPC has to solve a much larger (in terms of decision variables) optimization problem than the distributed LMPCs. We can also see that the evaluation time of the centralized LMPC is even larger than the sum of evaluation times of LMPC 1, LMPC 2, and LMPC 3 in the sequential distributed LMPC, and the times needed to solve the distributed LMPCs in both sequential and iterative distributed schemes are of the same order of magnitude.

In the following set of simulations, we compare the centralized LMPC and the two distributed LMPC schemes from a performance index point of view. In this set of simulations, the prediction horizon is $N = 1$. To carry out this comparison, the same initial condition and parameters were used for

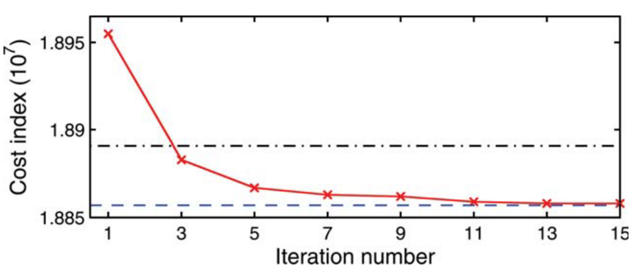


**Figure 6. Total performance cost along the closed-loop system trajectories of centralized LMPC (dashed line), sequential distributed LMPC (dash-dotted line), and the iterative distributed LMPC (solid line).**

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

the different control schemes and the total cost under each control scheme was computed as follows:

$$J = \sum_{i=0}^{M} x(t_i)^T Q_c x(t_i) + u_1(t_i)^T R_{c1} u_1(t_i) + u_2(t_i)^T R_{c2} u_2(t_i) + u_3(t_i)^T R_{c3} u_3(t_i)$$

where $t_0 = 0$ is the initial time of the simulations, $t_i = t_0 + i\Delta$ are the time instants taken into account, and $t_M = 1000$ s is the end of the simulations. Table 6 shows the total cost along the closed-loop system trajectories (trajectories I) under the different control schemes. For the iterative distributed MPC design, different maximum number of iterations, $c_{max}$, are used. From Table 6, we can see that in this set of simulations, the centralized LMPC gives the lowest performance cost, the sequential distributed LMPC gives lower cost than the iterative distributed LMPC when there is no iteration ($c_{max} = 1$). However, as the iteration number $c$ increases, the performance cost given by the iterative distributed LMPC decreases and converges to the cost of the one corresponding to the centralized LMPC. This point is also shown in Figure 6.

Note that the above set of simulations only represents one case of many possible cases. As we discussed in Remarks 11 and 12, there is no guaranteed convergence of the performance of distributed MPC to the performance of a centralized MPC and there is also no guaranteed superiority of the performance of one distributed LMPC scheme over the others. In the following, we show two sets of simulations to illustrate these points. In both sets of simulations, we chose different matrices $R_{c1}$ and $R_{c2}$, and all the other parameters ($Q_c$, $R_{c3}$, $\Delta$, $N$) remained the same as the previous set of simulations. In the first set of simulations, we picked $R_{c1} = \text{diag}([5 \times 10^{-5}\ 5 \times 10^{-5}\ 5 \times 10^{-5}])$, and $R_{c2} = \text{diag}([5 \times 10^{-5}\ 5 \times 10^{-5}])$. The total performance cost along the closed-loop system trajectories (trajectories II) under this simulation setting are shown in Table 7. From Table 7, we can see that the centralized LMPC provides a much lower cost than both the sequential and iterative distributed LMPCs. We can also see that as the number of iterations increases, the iterative distributed LMPC converges to a value which is different from the one obtained by the centralized LMPC. In the second set of simulations, we picked $R_{c1} = \text{diag}([1 \times 10^{-4}\ 1 \times 10^{-4}\ 1 \times 10^{-4}])$, $R_{c2} = \text{diag}([1 \times 10^{-4}\ 1 \times 10^{-4}])$ and the total performance cost along the closed-loop system

**Table 7. Total Performance Cost Along the Closed-Loop System Trajectories II**

| | | | | |
|---|---|---|---|---|
| | J ($\times 10^7$) | | | |
| Centralized | 5.052 | | | |
| Sequential | 7.039 | | | |
| $c_{max}$ | 1 | 3 | 5 | 6 |
| Iterative | 7.2286 | 7.2241 | 7.2240 | 7.2240 |

**Table 8. Total Performance Cost Along the Closed-Loop System Trajectories III**

| | | | |
|---|---|---|---|
| | | J ($\times 10^7$) | |
| Centralized | | 3.8564 | |
| Sequential | | 3.6755 | |
| $c_{max}$ | 1 | 3 | 4 |
| Iterative | 3.6663 | 3.6639 | 3.6639 |

trajectories (trajectories III) are shown in Table 8 from which we can see that the centralized LMPC provides a higher cost than both distributed LMPCs.

## Conclusions

In the present study, we presented two different architectures for distributed MPC for nonlinear process systems: sequential distributed MPC and iterative distributed MPC. In both architectures, the MPC controllers were designed via LMPC techniques. In the sequential distributed MPC architecture, the distributed LMPC controllers adopt a one-directional communication strategy and are evaluated in sequence and once at each sampling time; in the iterative distributed MPC architecture, the distributed LMPC controllers utilize a bi-directional communication strategy, are evaluated in parallel and iterate to improve closed-loop performance. Each LMPC controller in both architectures incorporates a suitable stability constraint which ensures that the state of the closed-loop system under the proposed distributed MPC architectures is ultimately bounded in an invariant set. Extensive simulations using a catalytic alkylation of benzene process example were carried out to compare the proposed distributed MPC architectures with existing centralized LMPC algorithms from computational time and closed-loop performance points of view.

## Acknowledgment

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

## Literature Cited

1. García CE, Prett DM, Morari M. Model predictive control: theory and practice–a survey. *Automatica.* 1989;25:335–348.
2. Rawlings JB. Tutorial overview of model predictive control. *IEEE Control Syst Mag.* 2000;20:38–52.
3. Camponogara E, Jia D, Krogh BH, Talukdar S. Distributed model predictive control. *IEEE Control Syst Mag.* 2002;22:44–52.
4. Rawlings JB, StewartBT. Coordinating Multiple Optimization-Based Controllers: New Opportunities and Chanllenges. In: Proceedings of 8th IFAC Symposium on Dynamics and Control of Process. Cancun, Mexico, 2007:19–28.
5. Scattolini R. Architectures for distributed and hierarchical model predictive control: a review. *J Process Control.* 2009;19:723–731.
6. Dunbar WB. Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Trans Autom Control.* 2007;52: 1249–1263.
7. Richards A, How JP. Robust distributed model predictive control. *Int J Control.* 2007;80:1517–1531.
8. 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.
9. 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. Seville, Spain, 2005:6680–6685.
10. Keviczky T, Borrelli F, Balas GJ. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica.* 2006;42:2105–2115.
11. Magni L, Scattolini R. Stabilizing decentralized model predictive control of nonlinear systems. *Automatica.* 2006;42:1231–1236.
12. 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.
13. Maestre JM, Muñoz de la Peña D, Camacho EF. A Distributed MPC Scheme with Low Communication Requirements. In: Proceedings of 2009 American Control Conference. Saint Louis, MO, 2009: 2797–2802.
14. Liu J, Muñoz de la Peña D, Christofides PD. Distributed model predictive control of nonlinear process systems. *AIChE J.* 2009;55: 1171–1184.
15. Massera JL. Contributions to stability theory. *Ann Math.* 1956;64: 182–206.
16. Lin Y, Sontag ED, Wang Y. A smooth converse Lyapunov theorem for robust stability. *SIAM J Control Optim.* 1996;34:124–160.
17. Christofides PD, El-Farra NH. *Control of Nonlinear and Hybrid Process Systems: Designs for Uncertainty, Constraints and Time-delays.* Berlin, Germany: Springer-Verlag, 2005.
18. Lin Y, Sontag ED. A universal formula for stabilization with bounded controls. *Syst Control Lett.* 1991;16:393–397.
19. Kokotovic P, Arcak M. Constructive nonlinear control: a historical perspective. *Automatica.* 2001;37:637–662.
20. El-Farra NH, Christofides PD. Integrating robustness, optimality and constraints in control of nonlinear processes. *Chem Eng Sci.* 2001; 56:1841–1868.
21. El-Farra NH, Christofides PD. Bounded robust control of constrained multivariable nonlinear processes. *Chem Eng Sci.* 2003; 58:3025–3047.
22. Mhaskar P, El-Farra NH, Christofides PD. Predictive control of switched nonlinear systems with scheduled mode transitions. *IEEE Trans Autom Control.* 2005;50:1670–1680.
23. 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.
24. Clarke F, Ledyaev Y, Sontag E. Asymtotic controllability implies feedback stabilization. *IEEE Trans Autom Control.* 1997;42:1394–1407.
25. 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.
26. Ganji H, Ahari JS, Farshi A, Kakavand M. Modelling and simulation of benzene alkylation process reactors for production of ethylbenzene. *Pet Coal.* 2004;46:55–63.
27. Lee WJ. Ethylbenzene dehydrogenation into styrene: kinetic modeling and reactor simulation. PhD thesis, Texas A&M University, College Station, TX, 2005.
28. Perego C, Ingallina P. Combining alkylation and transalkylation for alkylaromatic production. *Green Chem.* 2004;6:274–279.
29. You H, Long W, Pan Y. The mechanism and kinetics for the alkylation of benzene with ethylene. *Pet Sci Technol.* 2006;24:1079–1088.
30. Sontag E. A "universal" construction of Artstein's theorem on nonlinear stabilization. *Syst Control Lett.* 1989;13:117–123.
31. Wächter A, Biegler LT. On the implementation of primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Math Prog.* 2006;106:25–57.

*Manuscript received Aug. 18, 2009, and revision received Nov. 19, 2009.*