Distributed Lyapunov-Based Model Predictive Control with Neighbor-to-Neighbor Communication

Su Liu and Jinfeng Liu
Dept. of Chemical and Materials Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada

DOI 10.1002/aic.14579
Published online August 19, 2014 in Wiley Online Library (wileyonlinelibrary.com)

This work considers distributed predictive control of large-scale nonlinear systems with neighbor-to-neighbor communication. It fulfills the gap between the existing centralized Lyapunov-based model predictive control (LMPC) and the cooperative distributed LMPC and provides a balanced solution in terms of implementation complexity and achievable performance. This work focuses on a class of nonlinear systems with subsystems interacting with each other via their states. For each subsystem, an LMPC is designed based on the subsystem model and the LMPC only communicates with its neighbors. At a sampling time, a subsystem LMPC optimizes its future control input trajectory assuming that the states of its upstream neighbors remain the same as (or close to) their predicted state trajectories obtained at the previous sampling time. Both noniterative and iterative implementation algorithms are considered. The performance of the proposed designs is illustrated via a chemical process example. © 2014 American Institute of Chemical Engineers *AIChE J*, 60: 4124–4133, 2014

Keywords: nonlinear systems, predictive control, distributed control, chemical processes

Introduction

Large-scale, highly-integrated complex processes are increasingly used in modern process industries in order to get increased profits while meeting safety and environmental regulations. Model predictive control (MPC) has been widely used in the control of economically important units in these large-scale processes.[1] Typically, centralized MPC is not preferred in the control of large-scale processes due to organizational complexity, fault tolerance, as well as computational issues.[2–4] For plant-wide control of large-scale processes, the decentralized MPC is one of the solutions. To implement a decentralized MPC, a large-scale process is first decomposed into several subsystems such that the interactions between subsystems are minimized; and then a local MPC is designed for each subsystem generally without considering the interactions (e.g., Refs. 5–7). A key feature of decentralized MPC is that there is no or very little information exchange between subsystem MPCs. It is well known that the coupling between units in many processes such as integrated process networks with large recycles cannot be neglected[8–11] and that the plant-wide performance under decentralized control is limited because the interactions between subsystems in general are not considered.[12,13] These considerations motivate extensive studies in the past decade on distributed MPC (DMPC) in which subsystem MPCs communicate with each other to exchange information to coordinate their actions.

The available results on DMPC can be classified as noncooperative and cooperative algorithms based on the cost functions used in the subsystem MPCs. In a noncooperative DMPC algorithm, a subsystem MPC optimizes a local cost function. In Ref. 14, a noncooperative DMPC algorithm for a class of discrete-time linear systems was developed. In Ref. 15, a DMPC design was proposed for a class of dynamically decoupled nonlinear systems with coupled cost function. In Ref. 16, a DMPC for linear systems coupled through the state was considered. In Ref. 17, a DMPC algorithm based on dissipativity was developed for linear systems and a robust DMPC algorithm which takes into account model errors explicitly for linear systems was developed in Ref. 18.

In a cooperative DMPC algorithm, a global cost function is typically used in a subsystem MPC. The concept of cooperative DMPC was first introduced for linear systems in Ref. 19 and was further developed in Refs. 2,20. In this cooperative DMPC, a subsystem MPC needs to communicate with all the other subsystem MPCs every sampling time and the subsystem MPCs calculate their control actions in an iterative fashion. It was proven that the performance of the cooperative DMPC converges to the one of the corresponding centralized MPC as the iteration number increases. In another series of work,[21–23] sequential and iterative cooperative distributed Lyapunov-based MPC (LMPC) for general nonlinear systems were developed. In addition, algorithms based on agent negotiation were developed for linear systems coupled through the inputs.[24,25]

In another line of work, efforts have been devoted to the development of coordinated distributed MPC (CDMPC) algorithms. A feature of CDMPC is that there is a higher level coordinator which communicates with subsystem MPCs and coordinates their actions. In Refs. 26,27, a price-

Correspondence concerning this article should be addressed to J. Liu, jinfeng@ualberta.ca.

driven coordination algorithm was developed for general linear systems. The price-driven coordination was extended to handle nonlinear systems in Ref. 28. In Ref. 29, a sensitivity-driven coordination algorithm was developed for linear systems in which plant-wide coordination is achieved using a linear approximation of the objective functions of neighboring controllers within the objective function of each subsystem MPC. In Refs. 30,31, coordinated quasi-decentralized control schemes were developed for the control of large-scale systems to minimize the communication between subsystems and the coordinator while ensuring the closed-loop stability. More complete reviews of the results on DMPC from different perspectives can be found in Refs. 2–4.

In the above DMPC results, cooperative and coordinated DMPC algorithms may achieve the centralized control performance for linear systems which is achieved, however, at the cost of extensive communication between all the different subsystem MPCs or between the subsystem MPCs and the coordinator (i.e., all-to-all communication). When all-to-all communication is used, it could cause a high communication burden to the network especially when the system consists of a large number of subsystems. Moreover, the all-to-all communication may require significant higher maintenance efforts. Motivated by the above considerations, in this work, we develop DMPC algorithms for a class of nonlinear systems with neighbor-to-neighbor communication in the framework of Lyapunov-based MPC (LMPC). The neighbor-to-neighbor communication strategy has also been adopted in a few DMPC studies.[14,15,32,33] In Ref. 14, it assumes that the network of interactions between subsystems is sparse and it requires that controllers in a neighborhood work sequentially. In Ref. 15, a class of dynamically decoupled systems is considered. In Ref. 32, it considers nominal linear systems and uses a robust MPC approach to ensure the closed-loop stability in which at each sampling time, each subsystem sends to its neighbors information about its predicted future trajectory, and the subsystem's actual trajectory is restricted to be within a small neighborhood of the predicted trajectory. In Ref. 33, a strategy to reduce the communication for a class of linear systems in which subsystems are connected in sequence is developed. The main differences between the current work and Refs. 14,15,32,33 are in the following aspects: (1) we consider a class of general nonlinear systems and take bounded process disturbances into account; (2) an approach which imposes constraints on the change rates of the Lyapunov functions is developed to ensure the closed-loop stability; and (3) both noniterative and iterative implementation algorithms are considered.

The current work fulfills the gap between the existing centralized LMPC[34,35] and the cooperative distributed LMPC[21–23] and provides a balanced solution in terms of implementation complexity and achievable performance to plant-wide control of large-scale process systems. We note that it is not a trivial task to design distributed LMPC based on subsystem models and neighbor-to-neighbor communication. It requires different sets of fundamental assumptions and poses new challenges in the design of the subsystem MPCs as well as in the stability analysis. In the proposed designs, each subsystem controller is designed as an LMPC based on the subsystem model. A subsystem LMPC only needs to communicate with its neighbors to exchange subsystem state information. At a sampling time, a subsystem LMPC optimizes its local future control input trajectory assuming that the states of its upstream neighbors remain the same as or close to their predicted state trajectories obtained at the previous sampling time. Both noniterative and iterative implementation algorithms are considered and sufficient conditions for the closed-loop stability are derived. The effectiveness and applicability of the proposed designs are illustrated via the application to a chemical process example.

Preliminaries

Notation

Throughout this work, the operator $|\cdot|$ denotes the Euclidean norm of a scalar or a vector. A continuous function $\alpha : [0,a) \rightarrow [0,\infty)$ is said to belong to class $\mathcal{K}$ if it is strictly increasing and satisfies $\alpha(0)=0$. The symbol $\mathbb{I}$ denotes the set of all subsystem indices, which is defined as $\mathbb{I} : =\{1,\ldots,m\}$, where m is the total number of subsystems. The operator $\mathrm{diag}\,(x_1,\ldots,x_i,\ldots,x_n)$ denotes a diagonal matrix with its i-th diagonal element being x_i. The symbol "$\backslash$" denotes set substraction such that $\mathbb{A} \setminus \mathbb{B} : =\{x \in \mathbb{A}, x \notin \mathbb{B}\}$. The operator $\mathrm{cat}\,(\cdot)$ denotes the concatenation of two or more vectors such that $\mathrm{cat}\,(x_1,\ldots,x_n) : =[x_1^T,\ldots,x_n^T]^T$. Finally, the operator "×" when used in set operations denotes the Cartesian product such that: $\Omega_1 \times \Omega_2=\{(x_1,x_2) : x_1 \in \Omega_1 \text{ and } x_2 \in \Omega_2\}$.

System description

We consider a class of nonlinear systems composed of m interconnected subsystems with the i-th subsystem described by the following state-space model

$$\dot{x}_i(t)=f_i(X_i(t))+g_i(X_i(t))u_i(t)+k_i(X_i(t))w_i(t) \tag{1}$$

where $i \in \mathbb{I}, x_i \in \mathbb{R}^{n_{x_i}}$ denotes the state vector of subsystem i, $u_i \in \mathbb{R}^{n_{u_i}}$ denotes the control input vector of subsystem i, and $w_i \in \mathbb{R}^{n_{w_i}}$ denotes the disturbance vector of subsystem i. The state vector X_i is a concatenation of all the subsystem states involved in characterizing the dynamics of subsystem i and is defined as: $X_i : =\mathrm{cat}\,(x_l), l \in \mathbb{I}_i$, where $\mathbb{I}_i$ denotes the set of subsystem indices of which the corresponding subsystem states are involved in X_i. Subsystem l for $l \in \mathbb{I}_i \backslash \{i\}$ is called an upstream neighbor of subsystem i and accordingly, subsystem i is called a downstream neighbor of subsystem l.

The model of Eq. 1 also implies that the dynamics of the subsystems are coupled through their states only. The dynamics of the entire system is described as follows

$$\dot{x}(t)=f(x(t))+g(x(t))u(t)+k(x)w(t) \tag{2}$$

where $x \in \mathbb{R}^{n_x}, u \in \mathbb{R}^{n_u}$, and $w \in \mathbb{R}^{n_w}$ denote the state vector, control input vector, and disturbance vector of the entire system, respectively, with $x=[x_1^T,\ldots,x_i^T,\ldots,x_m^T]^T$, $u=[u_1^T,\ldots,u_i^T,\ldots,u_m^T]^T$, and $w=[w_1^T,\ldots,w_i^T,\ldots,w_m^T]^T$. The functions f, g, k are compositions of f_i, g_i, and k_i, with $f=[f_1^T,\ldots,f_i^T,\ldots,f_m^T]^T, g=\mathrm{diag}\,(g_1,\ldots,g_i,\ldots,g_m)$, and $k=\mathrm{diag}\,(k_1,\ldots,k_i,\ldots,k_m)$, The control input vector of subsystem i is restricted to be in a nonempty convex set $\mathbb{U}_i \subseteq \mathbb{R}^{n_{u_i}}, i \in \mathbb{I}$, such that

$$\mathbb{U}_i : =\{u_i \in \mathbb{R}^{n_{u_i}} : |u_i| \leq u_i^{\max}\} \tag{3}$$

where $u_i^{\max}, i \in \mathbb{I}$, is the magnitude of the input constraint. It is also assumed that the disturbance vector of subsystem i is bounded such that $w_i \in \mathbb{W}, i \in \mathbb{I}$, where

$$\mathbb{W}_i : =\{w_i \in \mathbb{R}^{n_{w_i}} : |w_i| \leq \theta_{w_i},\, \theta_{w_i} > 0\} \tag{4}$$

with $\theta_{w_i}, i \in \mathbb{I}$, being a known positive real number. Moreover, it is assumed that f_i, g_i, and k_i, $i \in \mathbb{I}$, are locally Lipschitz vector or matrix functions, and that $f_i(0)=0$ which implies that the origin is a steady state of the unforced nominal subsystem.

We assume that the states of the different subsystems are sampled synchronously and periodically at time instants $\{t_{k\geq 0}\}$ such that $t_k=t_0+k\Delta$ with $t_0=0$ being the initial time, Δ being a fixed time interval and k being positive integers.

Lyapunov-based controllers

In the proposed distributed LMPC designs, an auxiliary Lyapunov-based control law will be used in the design of each subsystem LMPC. It is assumed that for all $x_l \in \mathbb{D}_l$ where $l \in \mathbb{I}_i$ and $\mathbb{D}_l$ is a compact set containing the corresponding origin, there exists a nonlinear controller $u_i=h_i(X_i)$ that renders the corresponding origin of the nominal subsystem i, $i \in \mathbb{I}$, (i.e., subsystem (1) with $w_i(t) \equiv 0$ for all t) asymptotically stable while satisfying the input constraints. Note that in this assumption, the controller h_i depends on both the state of subsystem i and the states of subsystem i's upstream neighbors. It implies that a subsystem can be stabilized using its local control input as long as the states of its upstream neighbors are known.

Based on converse Lyapunov theorem,[36] the above assumption also implies that for each subsystem i, $i \in \mathbb{I}$, there exists a continuous differentiable control Lyapunov function $V_i(x_i)$ and $\mathcal{K}$ functions α_{ji}, $j=1,2,3,4$, that satisfy the following conditions

$$\begin{gathered}
\alpha_{1i}(|x_i|) \leq V_i(x_i) \leq \alpha_{2i}(|x_i|) \\
\frac{\partial V_i(x_i)}{\partial x_i}(f_i(X_i)+g_i(X_i)h_i(X_i)) \leq -\alpha_{3i}(|x_i|) \\
\left|\frac{\partial V_i(x_i)}{\partial x_i}\right| \leq \alpha_{4i}(|x_i|)
\end{gathered} \tag{5}$$

and $h_i(X_i) \in \mathbb{U}_i$ for all $x_l \in \mathbb{D}_l$, where $l \in \mathbb{I}_i$. In the remainder, we will refer to a level set of $V_i(x_i)$ in $\mathbb{D}_i, \Omega^i_{\rho_i}$, as the stability region of the closed-loop subsystem i under the control law $u_i=h_i(X_i)$. Specifically, the level set $\Omega^i_{\rho_i}$ is defined as: $\Omega^i_{\rho_i} := \{x_i \in \mathbb{D}_i : V_i(x_i) \leq \rho_i\}$. Accordingly, we define the stability region of the entire system as Ω, which is the Cartesian product of these stability regions as follows

$$\Omega=\Omega^1_{\rho_1}\times\Omega^2_{\rho_2}\ldots\times\Omega^m_{\rho_m} \tag{6}$$

Equation 6 indicates that if the state of each subsystem lies inside its stability region: $x_i \in \Omega^i_{\rho_i}, i \in \mathbb{I}$, then the state of the entire system model is within Ω: $x \in \Omega$.

By continuity and the locally Lipschitz properties of f_i, g_i, and k_i with respect to their arguments and taking into account the boundedness of the input vector u_i and disturbance vector w_i characterized in Eqs. 3 and 4, there exits a positive constant M_i for subsystem i, $i \in \mathbb{I}$, such that

$$|f_i(X_i)+g_i(X_i)u_i+k_i(X_i)w_i| \leq M_i \tag{7}$$

for all $x_l \in \Omega^l_{\rho_l}$ where $l \in \mathbb{I}_i$. In addition, by the continuous differentiable property of $V_i(x_i)$, and the locally Lipschitz properties of f_i, g_i, and k_i, there exist positive constants L_{f_i}, L_{g_i}, and L_{w_i} for subsystem i, $i \in \mathbb{I}$, such that:

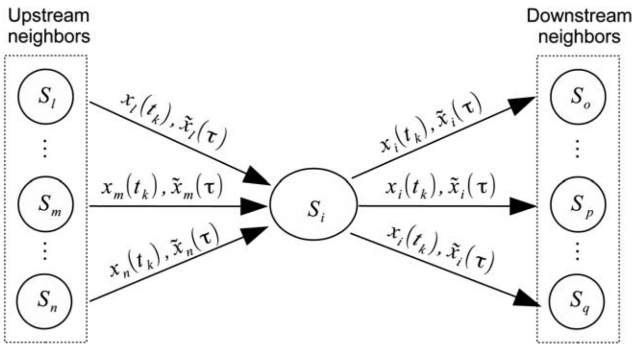
Upstream neighbors
Downstream neighbors
S_l
S_m
S_n
S_i
S_o
S_p
S_q
$x_l(t_k),\tilde{x}_l(\tau)$
$x_m(t_k),\tilde{x}_m(\tau)$
$x_n(t_k),\tilde{x}_n(\tau)$
$x_i(t_k),\tilde{x}_i(\tau)$
$x_i(t_k),\tilde{x}_i(\tau)$
$x_i(t_k),\tilde{x}_i(\tau)$

Figure 1. Information flow in the proposed DMPC design.

Subsystems *l*, *m*, *n* (i.e., S_l, S_m, S_n) are upstream neighbors of subsystem *i* (S_i); and subsystems *o*, *p*, *q* (i.e., S_o, S_p, S_q) are downstream neighbors of subsystem *i*. $\tilde{x}_l(\tau)$, $\tilde{x}_m(\tau), \tilde{x}_n(\tau)$ and $\tilde{x}_i(\tau)$ with $\tau \in [t_k, t_{k+N}]$ denote the predicted state trajectories of the corresponding subsystems.

$$\begin{gathered}
\left|\frac{\partial V_i(x_i)}{\partial x_i}f_i(X_i)-\frac{\partial V_i(x_i')}{\partial x_i}f_i(X_i')\right| \leq L_{f_i}|X_i-X_i'| \\
\left|\frac{\partial V_i(x_i)}{\partial x_i}g_i(X_i)-\frac{\partial V_i(x_i')}{\partial x_i}g_i(X_i')\right| \leq L_{g_i}|X_i-X_i'| \\
\left|\frac{\partial V_i(x_i)}{\partial x_i}k_i(X_i)\right| \leq L_{w_i}
\end{gathered} \tag{8}$$

for all $x_l, x_l' \in \Omega^l_{\rho_l}$ where $l \in \mathbb{I}_i$.

Remark 1. *The assumption of the existence of a set of nonlinear controllers $h_i(X_i), i \in \mathbb{I}$, means that a controller can stabilize its subsystem regardless of the states of its neighbors as long as their states are known to the controller. This assumption is different from the ones used in previous cooperative distributed LMPC work[21–23] in which a centralized nonlinear control law h(x) is required.*

Remark 2. *Note that the asymptotic stability assumption for the closed-loop subsystem i under the continuous control law $h_i(X_i)$ is based on the continuous availability of $X_i(t)$, The effect of sampling will be considered in the proposed distributed LMPC design and subsequent stability analysis.*

Proposed Distributed Predictive Control Algorithm: Noniterative Approach

In this section, we discuss the proposed distributed LMPC design for the noniterative case. To proceed, we first make the following statements/assumptions: (1) the controller associated with the i-th subsystem will be referred to as controller i; (2) controller i, $i \in \mathbb{I}$, has immediate and direct access to the measurements of x_i; (3) all controllers are evaluated synchronously at every sampling time; and (4) controller i, $i \in \mathbb{I}$, can communicate with its neighbors to exchange information. A schematic and the associated information flow of the proposed DMPC design is shown in Figure 1. At a sampling instant t_k, subsystem i receives the state measurements as well as predicted future state trajectories from its upstream neighbors and sends its state measurement and predicted future state trajectory to its downstream neighbors.

Implementation strategy

The detailed implementation algorithm of the proposed noniterative distributed LMPC is as follows:

Algorithm 1. *Noniterative implementation algorithm at sampling instant t_k*

1. *Controller i, $i \in \mathbb{I}$, receives $x_i(t_k)$, and then sends $x_i(t_k)$ and the predicted state trajectory obtained at the previous time instant $t_{k-1}, \tilde{x}_i(\tau), \tau \in [t_k, t_{k+N}]$, to its downstream neighbors. Meanwhile, it receives the subsystem states $x_l(t_k), l \in \mathbb{I}_i \backslash \{i\}$, and the predicted state obtained at $t_{k-1}, \tilde{x}_l(\tau), \tau \in [t_k, t_{k+N}], l \in \mathbb{I}_i \backslash \{i\}$, via the communication network from its upstream neighbors.*

2. *Controller i, $i \in \mathbb{I}$, evaluates its future input trajectory based on measurement $X_i(t_k)$ and predicted subsystem state trajectories $\tilde{X}_i(\tau)$. After the evaluation, controller i, $i \in \mathbb{I}$, sends its first control input to its actuator.*

In Algorithm 1, the predicted state trajectories sent from controller i, $i \in \mathbb{I}$, to its downstream neighbors at sampling time t_k is obtained from the evaluation of controller i at the previous sampling instant t_{k-1}. In the evaluation of a subsystem controller, it assumes that the states of its upstream neighbors will evolute as the predicted state trajectories sent to it. Regarding the startup of Algorithm 1 (i.e., at time instant t_0), each subsystem could assume zero state for other subsystems; that is, subsystem i, $i \in \mathbb{I}$, assumes that $\tilde{x}_l(t)=0, l \in \mathbb{I}_i \backslash \{i\}$, for $t \in [t_0, t_N]$. This initialization may affect the control performance but does not affect the stability of the proposed design which will be made clear in the following subsections.

Remark 3. *Compared with the cooperative DMPC, the implementation strategy outlined in Algorithm 1 significantly reduces the network communication in the following aspects: (1) the communication only takes place between neighbors; and (2) only one-sided communication is needed from an upstream subsystem to its downstream neighbors. It is expected that the reduction of the amount of communication in the proposed distributed LMPC tends to be more significant for systems with sparsely connected subsystems. However, we note that the reduction of the amount of communication might lead to the loss of the plant-wide control performance. We will design the proposed distributed LMPC to ensure the closed-loop system stability while demonstrating its performance via simulations which is an approach that has been widely used in studies of DMPC of nonlinear systems.*

Subsystem LMPC formulation

In the proposed distributed LMPC, a subsystem LMPC is designed based on its corresponding subsystem nominal model. The objective of the proposed design is to collaboratively stabilize the entire system at the origin using neighbor-to-neighbor communication. At a sampling time, a subsystem LMPC uses its subsystem model to optimize its future control input trajectory assuming that the future state trajectories of its upstream neighbors remain the same as (or in a small neighborhood of) the predictions. In LMPC i, the nonlinear controller $h_i(X_i)$ is utilized to construct a stability constraint. These stability constraints in the m LMPCs together ensure the closed-loop stability of the entire system. Specifically, the proposed design of LMPC i at t_k is formulated as follows

$$u_i^*(\tau|t_k)=\arg \min_{u_i(\tau)\in S(\Delta)} \int_{t_k}^{t_{k+N}} [\tilde{x}_i(\tau)^T Q_i \tilde{x}_i(\tau)+u_i(\tau)^T R_i u_i(\tau)]\mathrm{d}\tau \tag{9a}$$

$$\text{s.t. } \dot{\tilde{x}}_i(\tau)=f_i(\tilde{X}_i(\tau))+g_i(\tilde{X}_i(\tau))u_i(\tau) \tag{9b}$$

$$\tilde{X}_i(t_k)=X_i(t_k) \tag{9c}$$

$$u_i(\tau) \in \mathbb{U}_i \tag{9d}$$

$$\frac{\partial V_i(x_i(t_k))}{\partial x_i} g_i(X_i(t_k))u_i(t_k) \leq \frac{\partial V_i(x_i(t_k))}{\partial x_i} g_i(X_i(t_k))h_i(X_i(t_k)) \tag{9e}$$

where $u_i^*(\tau|t_k)$ denotes the optimal input sequence to the optimization problem of Eq. 9, $S(\Delta)$ denotes the family of continuous piecewise functions with sampling time Δ, N is the prediction horizon, Q_i and R_i are positive definite weighting matrices, $\tilde{x}_i(\tau), \tau \in [t_k, t_{k+N}]$, is the predicted state trajectory of the nominal subsystem i, and $\tilde{X}_i(\tau)$ is the concatenated vector of $\tilde{x}_i(\tau), l \in \mathbb{I}_i$. We also denote the predicted subsystem state trajectory obtained in Eq. 9 corresponding to the optimal input sequence $u_i^*(\tau|t_k)$ as $\tilde{x}_i^*(\tau|t_k)$ which will be used to describe the information exchanged between subsystems.

In the above optimization, Eq. 9a is the objective function involving the state and input vector of subsystem i. Equation (9b) is the nominal model of subsystem i, in which the predicted state trajectories of the upstream neighbors, $x_l(\tau), l \in \mathbb{I}_i \backslash \{i\}, \tau \in [t_k, t_{k+N}]$, are obtained via communication. Equation (9c) specifies the initial condition at time t_k in which $x_i(t_k)$ is directly available and $x_l(t_k), l \in \mathbb{I}_i \backslash \{i\}$, are obtained via communication. Equation (9d) is the constraint on control input u_i. Equation (9e) is the stability constraint based on the nonlinear controller $h_i(X_i)$ as well as the associated Lyapunov function V_i. The manipulated inputs of the closed-loop system under the proposed distributed LMPC at time t_k are defined as follows

$$u_i(t)=u_i^*(t|t_k), \;\; t \in [t_k, t_{k+1}), \;\; i \in \mathbb{I}. \tag{10}$$

After solving the optimization problem of Eq. 9, the predicted state trajectory of subsystem i is obtained as follows

$$\tilde{x}_i(\tau)=\tilde{x}_i^*(\tau|t_k), \quad \tau \in [t_{k+1}, t_{k+N}] \tag{11a}$$

$$\tilde{x}_i(\tau)=\tilde{x}_i^*(t_{k+N}|t_k), \quad \tau \in (t_{k+N}, t_{k+N+1}] \tag{11b}$$

This predicted trajectory $\tilde{x}_i(\tau), \;\; \tau \in [t_{k+1}, t_{k+N+1}]$, together with the next measurement $x_i(t_{k+1})$, will be transmitted to the downstream neighbors of subsystem i at sampling instant t_{k+1}, as elaborated in Step 1 of Algorithm 1. In Eq. 11b, we assume that the predicted state trajectory remains constant as its last evaluated value when the prediction time goes one sampling interval beyond the optimization window. Note that this assumption may have a minor effect on the control performance but does not affect the stability property of the proposed algorithm.

Stability analysis

The proposed LMPC design of Eq. 9 is based on the continuous Lyapunov-based controller $h_i(X_i)$ whose stability properties are characterized in Eq. 5. The proposed LMPC inherits the stability properties of the Lyapunov-based controller when it is implemented in a sample-and-hold fashion. In this subsection, we provide a set of sufficient conditions for the closed-loop stability of the proposed noniterative distributed LMPC algorithm in which the effects of sampling are explicitly characterized.

To begin with, let us consider a set of scalars ρ_{s_i} with $i \in \mathbb{I}$, such that $\rho_{s_i} < \rho_i, i \in \mathbb{I}$, and define $\rho_i^*, i \in \mathbb{I}$, as follows

$$\rho_i^* := \max\{V_i(x_i(t+\Delta)) : V_i(x_i(t)) \le \rho_{s_i}\} \tag{12}$$

Note that ρ_i^* is defined as the biggest possible value that V_i may achieve in one sampling time (i.e., Δ) when the initial value is smaller than or equal to ρ_{s_i}. Based on $\rho_i^*, i \in \mathbb{I}$, let us further define Ω^* as follows

$$\Omega^* := \Omega^1_{\rho_1^*} \times \Omega^2_{\rho_2^*} \times \ldots \times \Omega^m_{\rho_m^*} \tag{13}$$

Finally, we define $M_i^X, i \in \mathbb{I}$, such that

$$M_i^X := \sqrt{\sum\nolimits_{l \in \mathbb{I}_i} M_l^2} \tag{14}$$

where M_l, $l \in \mathbb{I}_i$, are the positive constants defined in Eq. 7.

Theorem 1 below provides a set of sufficient conditions under which the closed-loop stability of the proposed noniterative distributed LMPC is guaranteed.

Theorem 1. *Consider system (2) in closed-loop with each subsystem controller designed as in Eq. 9 based on $h_i(X_i), i \in \mathbb{I}$, that satisfy the conditions in Eq. 5 with class $\mathcal{K}$ functions α_{ji}, $j=1,2,3,4$ and $i \in \mathbb{I}$. If the distributed controllers are implemented following Algorithm 1 and there exist $\epsilon_i > 0, \Delta > 0$, and $\rho_i > \rho_{s_i} > 0$ that satisfy the following inequalities*

$$-\alpha_{3i}(\alpha_{2i}^{-1}(\rho_{s_i})) + (L_{f_i} + L_{g_i} u_i^{max}) M_i^X \Delta + L_{w_i}\theta_{w_i} \le -\epsilon_i/\Delta \tag{15}$$

for $i \in \mathbb{I}$ with L_{f_i}, L_{g_i}, and L_{w_i} defined in Eq. 8, and if $x_i(t_0) \in \Omega^i_{\rho_i}$ and $\rho_i^* < \rho_i$, then the closed-loop subsystem state $x_i(t) \in \Omega^i_{\rho_i}$ for all $t \ge t_0$ and is ultimately bounded in $\Omega^i_{\rho_i^*}$. This also implies that the entire system state is always maintained in Ω and is ultimately bounded in Ω^*.

Proof. The proof is divided into two parts. In the first part, we show that the optimization problem of Eq. 9 is feasible for all $x_i(t_k) \in \Omega^i_{\rho_i}, i \in \mathbb{I}$. In the second part, we prove that if the conditions in Eq. 15 hold and if $x(t_0) \in \Omega$, then $x(t)$ is maintained in Ω and is ultimately bounded in Ω^*.

Part 1. To prove the feasibility of the optimization problem of Eq. 9, we only have to find an input sequence that satisfies the constraints of Eqs. 9d and 9e. Note that the constraint of Eq. 9e is imposed only on $u_i(t_k)$ and that $u_i(t_k)=h_i(X_i(t_k))$ satisfies the inequality of Eq. 9e with an equal sign. Note also that $u_i(t_k)=h_i(X_i(t_k))$ does not violate Eq. 9d which is an assumed property for the Lyapunov-based controller $h_i(X_i)$ for all $x_l(t_k) \in \Omega^l_{\rho_l}, l \in \mathbb{I}_i$. Therefore, an input sequence that satisfies the constraints of the optimization problem of Eq. 9 can be found with $u_i(t_k)=h_i(X_i(t_k))$, and $u_i(\tau), \tau=t_{k+1}, \ldots, t_{k+N-1}$, being any value in the convex set $\mathbb{U}_i$.

Part 2. If the initial states of all the subsystems at time instant t_k are inside their corresponding stability regions: $x_i(t_k) \in \Omega^i_{\rho_i}, i \in \mathbb{I}$. From Eq. 9e and the conditions in Eq. 5, the following inequality can be obtained

$$\begin{aligned}&\frac{\partial V_i}{\partial x_i}(f_i(X_i(t_k)) + g_i(X_i(t_k))u_i^*(t_k|t_k))\\ &\quad \le \frac{\partial V_i}{\partial x_i}(f_i(X_i(t_k)) + g_i(X_i(t_k))h_i(X_i(t_k))) \le -\alpha_{3i}(|x_i(t_k)|)\end{aligned} \tag{16}$$

The time derivative of the Lyapunov function $V_i(x_i)$ of subsystem i, $i \in \mathbb{I}$, in $t \in [t_k, t_{k+1})$ can be evaluated as follows

$$\dot{V}_i(x_i(t)) = \frac{\partial V_i}{\partial x_i}(f_i(X_i(t)) + g_i(X_i(t))u_i^*(t_k|t_k) + k_i(x_i(t))w(t)) \tag{17}$$

Combining Eqs. 16 and 17 yields the following inequality

$$\begin{aligned}\dot{V}_i(x_i(t)) &\le -\alpha_{3i}(|x_i(t_k)|)\\ &+ \frac{\partial V_i}{\partial x_i}(f_i(X_i(t)) + g_i(X_i(t))u_i^*(t_k|t_k) + k_i(x_i(t))w(t))\\ &- \frac{\partial V_i}{\partial x_i}(f_i(X_i(t_k)) + g_i(X_i(t_k))u_i^*(t_k|t_k))\end{aligned} \tag{18}$$

Applying the conditions in Eqs. 5 and 8, the following inequality holds for all $x_i(t_k) \in \Omega_{\rho_i} \backslash \Omega_{\rho_{s_i}}$ and $x_l(t_k) \in \Omega_{\rho_l}$ where $l \in \mathbb{I}_i \backslash \{i\}$

$$\dot{V}_i(x_i(t)) \le -\alpha_{3i}(\alpha_{2i}^{-1}(\rho_{s_i})) + (L_{f_i} + L_{g_i}u_i^{\max})|X_i(t) - X_i(t_k)| + L_{w_i}\theta_{w_i} \tag{19}$$

Taking into account Eq. 7 and the continuity property of $x_i(t)$, the following bound for the state vector of each subsystem i, $i \in \mathbb{I}$, within every sampling period $t \in [t_k, t_{k+1})$ can be established

$$|x_i(t) - x_i(t_k)| \le M_i\Delta \tag{20}$$

And the bound for the concatenated state vector $X_i(t)$ within every sampling period $t \in [t_k, t_{k+1})$ can be established

$$|X_i(t) - X_i(t_k)| = \sqrt{\sum_{l \in \mathbb{I}_i} |x_l(t) - x_l(t_k)|^2} \le \sqrt{\sum_{l \in \mathbb{I}_i} (M_l\Delta)^2} = M_i^X\Delta \tag{21}$$

Substituting Eq. 21 into Eq. 19, we obtain the following inequality

$$\dot{V}_i(x_i(t)) \le -\alpha_{3i}(\alpha_{2i}^{-1}(\rho_{s_i})) + (L_{f_i} + L_{g_i}u_i^{max})M_i^X\Delta + L_{w_i}\theta_{w_i} \tag{22}$$

If the condition in Eq. 15 holds, then for all $x_i(t_k) \in \Omega_{\rho_i} \backslash \Omega_{\rho_{s_i}}$ and $x_l(t_k) \in \Omega_{\rho_l}$ where $l \in \mathbb{I}_i \backslash \{i\}$, there exists $\epsilon_i > 0$ such that

$$\dot{V}_i(x_i(t)) \le -\epsilon_i/\Delta \tag{23}$$

for $t \in [t_k, t_{k+1})$. The above inequality indicates that $V_i(x_i(t))$ remains decreasing for $t \in [t_k, t_{k+1})$. Integrating $V_i(x_i(t))$ from $t = t_k$ to t_{k+1}, the following inequality can be obtained from Eq. 23

$$V_i(x_i(t_{k+1})) \le V_i(x_i(t_k)) - \epsilon_i \tag{24}$$

Thus, for all $x_i(t_k) \in \Omega_{\rho_i} \backslash \Omega_{\rho_{s_i}}$ and $x_l(t_k) \in \Omega_{\rho_l}$ where $l \in \mathbb{I}_i \backslash \{i\}$, the state $x_i(t)$ converges to $\Omega_{\rho_{s_i}}$ in a finite number of sampling times without leaving the stability region Ω_{ρ_i}. Once $x_i(t)$ converges to $\Omega_{\rho_i^*}$, it remains in $\Omega_{\rho_i^*}$ for all time due to the definition of $\Omega_{\rho_i^*}$.

Because the states of all subsystems $x_i(t), i \in \mathbb{I}$, are maintained in their corresponding stability region Ω_{ρ_i} for all $t > t_0$ and are ultimately bounded in $\Omega_{\rho_i^*}$, the entire system state x is maintained in Ω for all time and is ultimately bounded in Ω^*. ■

Iterative Implementation

In the previous section, a noniterative subsystem-based distributed LMPC implementation strategy is discussed. At each sampling time, a subsystem MPC assumes that the states of its upstream neighbors remain the same as or close to the predicted values at a previous sampling time. The performance of the proposed distributed LMPC may be improved by allowing the distributed controllers to exchange information and recalculate their actions iteratively every sampling time. In this section, we discuss the implementation of the proposed subsystem-based distributed LMPC in which an iterative procedure is carried out in the controller evaluation.

Implementation strategy

The implementation strategy of the proposed iterative subsystem-based distributed LMPC at a time instant t_k is as follows:

Algorithm 2. *Iterative implementation algorithm at* t_k

1. *Controller i,* $i \in \mathbb{I}$*, receives and sends* $x_i(t_k)$ *to its downstream neighbors. Meanwhile, it receives the subsystem states* $x_l(t_k), l \in \mathbb{I}_i \backslash \{i\}$*, via the communication network from its upstream neighbors.*

2. *(Iterative controller evaluation)*

2.1. *At iteration c* ($c \geq 1$)*, Controller i,* $i \in \mathbb{I}$*, evaluates its future input trajectory based on the state trajectories of its upstream neighbors evaluated at iteration c – 1 and* $X_i(t_k)$*.*

2.2 *Controller i,* $i \in \mathbb{I}$*, sends the predicted state trajectory* $\tilde{x}_i^c(\tau|t_k), \tau \in [t_k, t_{k+N}]$ *to its downstream neighbors, and receives the predicted state trajectories of its upstream neighbors* $\tilde{x}_l^c(\tau|t_k), l \in \mathbb{I}_i, \tau \in [t_k, t_{k+N}]$*.*

In Algorithm 2, controllers evaluate their control inputs and communicate their predicted future state trajectories iteratively before the control inputs evaluated at the last iteration are implemented. At the initial iteration ($c=1$), upstream neighboring subsystem state trajectories are initialized by their predicted future trajectories at the previous sampling instant t_{k-1}. Regarding the termination conditions, there can be different choices. For example, the number of iterations c may be restricted to be smaller than a maximum iteration number $c \leq c_{\max}$ 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. The startup of Algorithm 2 may follow the same procedure as Algorithm 1.

DMPC formulation

The proposed design of LMPC i, $i \in \mathbb{I}$, at a sampling time t_k, iteration c, is formulated as the following optimization problem

$$u_i^{*,c}(\tau|t_k)=\arg \min_{u_i(\tau)\in S(\Delta)} \int_{t_k}^{t_{k+N}} [\tilde{x}_i(\tau)^T Q_i \tilde{x}_i(\tau)+u_i(\tau)^T R_i u_i(\tau)]\mathrm{d}\tau \tag{25a}$$

$$\text{s.t. } \dot{\tilde{x}}_i(\tau)=f_i(\tilde{X}_i^{c-1}(\tau))+g_i(\tilde{X}_i^{c-1}(\tau))u_i(\tau) \tag{25b}$$

$$\tilde{X}_i^{c-1}(t_k)=X_i(t_k) \tag{25c}$$

$$u_i(\tau) \in \mathbb{U}_i \tag{25d}$$

$$\frac{\partial V_i(x_i(t_k))}{\partial x_i} g_i(X_i(t_k))u_i(t_k) \leq \frac{\partial V_i(x_i(t_k))}{\partial x_i} g_i(X_i(t_k))h_i(X_i(t_k)) \tag{25e}$$

where $u_i^{*,c}(\tau|t_k)$ denotes the optimal solution to the optimization problem of Eq. 25, $\tilde{x}_i$ is the predicted state trajectory of the nominal subsystem i evaluated at iteration c, $\tilde{X}_i^{c-1}(\tau)$ denotes the concatenated vector of $\tilde{x}_i(\tau)$ and $\tilde{x}_l^{c-1}(\tau), l \in \mathbb{I}_i \backslash \{i\}$, where $\tilde{x}_l^{c-1}(\tau)$ denotes the state trajectories of upstream neighbors of subsystem i evaluated at iteration $c - 1$. The manipulated inputs of the closed-loop system under the iterative implementation are defined as follows

$$u_i(t)=u_i^{*,c_t}(t|t_k), \ t \in [t_k, t_{k+1}), \ i \in \mathbb{I} \tag{26}$$

where c_t denotes the iteration times when the iteration is terminated. After the iterative controller evaluation is terminated, the corresponding predicted state trajectory of subsystem i is transmitted to its downstream neighbors.

Stability analysis

The stability properties established for the noniterative algorithm are maintained for the iterative implementation because of the stability constraint of Eq. 25e. The stability properties of the proposed iterative distributed LMPC design are summarized in Theorem 2 below

Theorem 2. *Consider system (2) in closed-loop with each subsystem controller designed as in Eq. 25 based on* $h_i(X_i), i \in \mathbb{I}$*, that satisfy the conditions in Eq. 5 with class* $\mathcal{K}$ *functions* α_{ji}*,* $j=1,2,3,4$ *and* $i \in \mathbb{I}$*. If the distributed controllers are implemented following Algorithm 2 and there exist* $\epsilon_i > 0, \Delta > 0$*, and* $\rho_i > \rho_{s_i} > 0$ *that satisfy the constraint of Eq. 15 for* $i \in \mathbb{I}$ *with* L_{f_i}, L_{g_i}*, and* L_{w_i} *defined in Eq. 8, and if* $x_i(t_0) \in \Omega^i_{\rho_i}$ *and* $\rho_i^* < \rho_i$*, then the closed-loop subsystem state* $x_i(t) \in \Omega^i_{\rho_i}$ *for all* $t \geq t_0$ *and is ultimately bounded in* $\Omega^i_{\rho_i^*}$*. This also implies that the entire system state is always maintained in* Ω *and is ultimately bounded in* Ω^**.*

Proof. Similar to the proof of Theorem 1, the proof of Theorem 2 is divided into two parts. In the first part, we show that the optimization problem of Eq. 25 is feasible for all $x_i(t_k) \in \Omega^i_{\rho_i}, i \in \mathbb{I}$. In the second part, we prove that if the conditions in Eq. 15 hold and if $x(t_0) \in \Omega$, then $x(t)$ is maintained in Ω and is ultimately bounded in Ω^*.

Part 1. To prove the feasibility, we only have to find an input sequence that satisfies the constraints of Eqs. 25d and 25e. Because the input bound of Eq. 25d and the stability constraint of Eq. 25e remain the same as in the noniterative algorithm at every iteration, they can be satisfied with the same type of input sequence with $u_i(t_k)=h_i(X_i(t_k))$, and $u_i(\tau), \tau=t_{k+1}, \ldots, t_{k+N-1}$, being any value in the convex set $\mathbb{U}_i$.

Part 2. If the initial states of all the subsystems at t_k are inside their corresponding stability regions, $x_i(t_k) \in \Omega^i_{\rho_i}, i \in \mathbb{I}$. From Eq. 25e and the condition in Eq. 5, the following inequality can be obtained for the optimal control input evaluated at iteration c

$$\begin{aligned} &\frac{\partial V_i}{\partial x_i}(f_i(X_i(t_k))+g_i(X_i(t_k))u_i^{*,c}(t_k|t_k)) \\ &\quad \leq \frac{\partial V_i}{\partial x_i}(f_i(X_i(t_k))+g_i(X_i(t_k))h_i(X_i(t_k))) \leq -\alpha_{3i}(|x_i(t_k)|) \end{aligned} \tag{27}$$

Following the similar arguments as in the proof of Theorem 1, it can be shown that if the conditions in Eq. 15 hold, then the states of all subsystems $x_i(t), i \in \mathbb{I}$, are maintained in their corresponding stability region Ω_{ρ_i} for all $t > t_0$ and ultimately bounded in $\Omega_{\rho_i^*}$ if $u_i^{*,c}(t_k|t_k)$ is implemented, which means that the entire system state x is kept in Ω all the time and is ultimately bounded in Ω^*. ■

Application to A Reactor-Separator Process

Process description

In this section, we consider a reactor-separator process which consists of two continuously stirred tank reactors

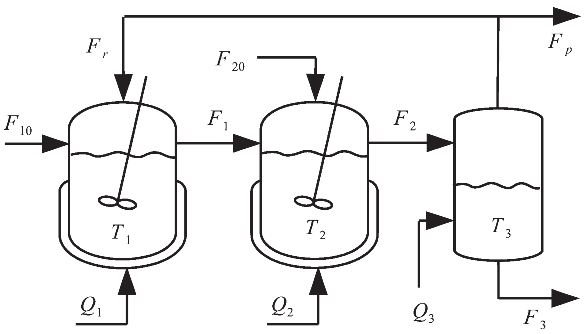

Figure 2. Reactor-separator process.

(CSTRs) and a flash tank separator shown in Figure 2. In this process, two elementary reactions take place in the two CSTRs: $A \rightarrow B$ and $A \rightarrow C$. Pure reactant A is fed into CSTR 1 at flow rate F_{10}. The effluent of CSTR 1 is fed into CSTR 2 at flow rate F_1 along with another fresh feed of reactant A at flow rate F_{20}. The effluent of CSTR 2 is fed into the flash tank at flow rate F_2. The bottom product of the flash tank is removed at flow rate F_3. The overhead vapor from the flash tank is condensed and recycled to CSTR 1 at flow rate F_r, with a small portion purged at flow rate F_p. A jacket is equipped with each vessel to provide/remove heat. The dynamic behavior of the process is modeled by 12 nonlinear ordinary differential equations based on mass and energy balances under standard modeling assumptions. The detailed mathematical model can be found in Ref. 37. The state vector of the process includes the temperatures and species concentrations in the three vessels and is defined as follows

$$x=[T_1, C_{A1}, C_{B1}, C_{C1}, T_2, C_{A2}, C_{B2}, C_{C2}, T_3, C_{A3}, C_{B3}, C_{C3}]^T$$

where T_i, $i=1, 2, 3$, denote the temperature (K) of the three vessels and C_{Ai}, C_{Bi}, C_{Ci}, $i=1, 2, 3$, denote the concentrations ($kmol/m^3$) of A, B, C in the three vessels, respectively. The desired operating point x_s is an unstable steady state of the process and is

$$x_s=[369.5, 3.318, 0.172, 0.042, 435.3, 2.751, 0.466, \\ 0.111, 435.3, 2.882, 0.497, 0.120]^T.$$

The objective is to steer the process state to x_s and maintain it at x_s by manipulating the heat input/removal to/from the vessels (i.e., Q_1, Q_2, Q_3).

Subsystem LMPC design

The process is divided into three subsystems with respect to the three vessels. The state vector of subsystem i, $i=1, 2, 3$, is defined as

$$x_i=[T_i, C_{Ai}, C_{Bi}, C_{Ci}]^T \tag{28}$$

It can be verified based on the process model (see Ref. 37) that the dynamics of the three subsystems can be described in the form of Eq. 1 with $X_1=[x_1^T, x_3^T]^T, X_2=[x_1^T, x_2^T]^T$, and $X_3=[x_2^T, x_3^T]^T$. The control input of each subsystem is the heat input/removal to/from each vessel: $u_i=Q_i$, $i=1, 2, 3$, ($kJ\ /\ h$). We design $u_i=h_i(X_i)$, $i=1, 2, 3$, using feedback linearization technique. Specifically, $h_i(X_i)$, $i=1, 2, 3$, takes the following form

$$h_i(X_i(t))=g_i(X_i(t))^{-1}(-f_i(X_i(t))-ax_i(t)) \tag{29}$$

where $a=2$. A quadratic Lyapunov function $V_i(x)=x_i^T P_i x_i$ is used with $P_i=\text{diag}(10, 10^4, 10^4, 10^4)$. The parameters of the subsystem LMPC i, $i=1, 2, 3$, are chosen as follows: the sampling time is $\Delta=0.025h$; the prediction horizon is $N=5$; the weighting matrices are $Q_i=\text{diag}(10, 10^4, 10^4, 10^4)$ and $R_i=10^{-7}$; the control input is bounded by $|u_i| \leq 10^6 kJ/h$.

Simulation results

In this subsection, different LMPC implementations are compared with the proposed subsystem-based distributed LMPC algorithms to provide a comprehensive view of the performance of the proposed designs. Specifically, the proposed subsystem-based distributed LMPC algorithms will be compared with a centralized LMPC,[34,35] a decentralized LMPC in which the three controllers do not communicate and a controller assumes that the other two subsystems are at their steady states, and an iterative cooperative distributed LMPC.[21–23] All LMPC implementations use the same controller parameters and Lyapunov-based controllers.

The same initial condition and disturbance sequences will be used for each run under different LMPC implementations. The initial system state is

$$x_0=[356.9, 3.227, 0.030, 0.015, 446.5, 2.696, 0.465, \\ 0.125, 446.5, 2.821, 0.530, 0.139]^T$$

In the case where process disturbances are present, bounded random disturbances are added to the right hand side of the dynamic equations of the process. The random disturbances added to the dynamics of the temperatures are generated as normal distributed noises with zero mean and standard deviation 10 K/s in the range [−30K/s, 30K/s]. The random disturbances added to the dynamics of concentrations are generated as normal distributed noises with zero mean and standard deviation 0.01kmol/m^{3}s in the range

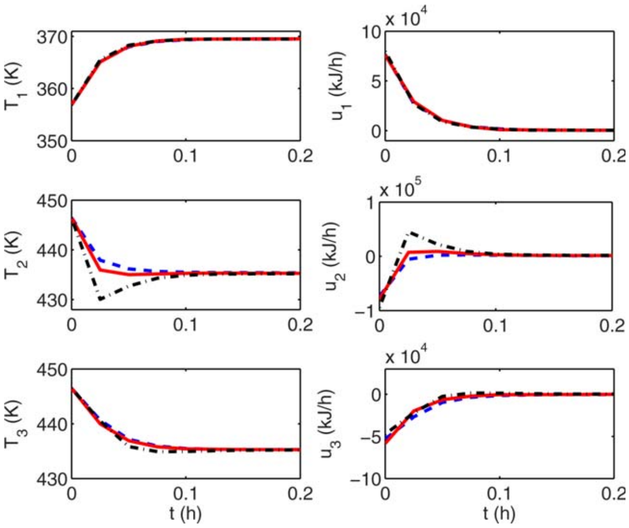

Figure 3. Temperature and input trajectories of the nominal closed-loop system under the centralized LMPC (solid line), the decentralized LMPC (dash dotted line), and the proposed noniterative distributed LMPC (dashed line).

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

Table 1. Nominal Performance Indices and CPU Times of the Centralized LMPC, the Proposed Non-Iterative Distributed LMPC and the Decentralized LMPC

	Centralized LMPC	Non-Iterative Distributed LMPC	Decentralized LMPC
J	276.5	309.7	346.4
CPU time	12.9 s	1.7 s	1.9 s

$[-0.03\text{kmol/m}^3\text{s},\ 0.03\text{kmol/m}^3\text{s}]$. It is assumed that the disturbances remain constant during every sampling period.

In the first set of simulations, we compare the proposed noniterative distributed LMPC implemented following Algorithm 1 with two typical noniterative MPC configurations: the centralized LMPC and the decentralized LMPC. The centralized LMPC optimizes the entire system objective function based on the centralized system model of Eq. 2. In decentralized LMPC, each LMPC $i,\ i \in \mathbb{I}$, optimizes a local objective function based on the subsystem model of Eq. 1, with the states of its upstream neighbors $x_l,\ l \in \mathbb{I}_i \backslash \{i\}$, being fixed at their steady-state values.

We first compare the three noniterative control configurations for the disturbance free case. The temperature and input trajectories of the nominal closed-loop system under the three control configurations are shown in Figure 3. In addition, let us define the performance index as follows

$$J=\frac{1}{F}\sum_{t_k=t_1}^{t_F}\left[x(t_k)^T Q x(t_k)+u(t_k)^T R u(t_k)\right] \tag{30}$$

where $t_F=0.2h$ is the simulation time, $x(t_k)$ is the entire system state at t_k, $Q=\text{diag}(Q_1,Q_2,Q_3)$, and $R=\text{diag}(R_1,R_2,R_3)$. The performance indices and average CPU times consumed in a single evaluation by the three control configurations are shown in Table 1. The results in Figure 3 and Table 1 indicate that the proposed noniterative distributed LMPC algorithm obtains an improved control performance over decentralized LMPC while maintaining the same computational complexity. Note that the improved performance is due to communication of state measurements and predicted state trajectories between neighboring subsystems.

Next we compare the three noniterative implementations with process disturbances. Ten Runs with different sequences of disturbances of each of the three noniterative control configurations are carried out. The performance indices and CPU times of these runs are shown in Table 2. The results agree with the conclusions obtained in the nominal case.

Table 2. Performance Indices (*J*) and CPU Times (*CPU*) of the Centralized LMPC, the Proposed Non-Iterative Distributed LMPC and the Decentralized MPC in 10 Runs

Run #	Centralized (J, CPU)	Proposed (J, CPU)	Decentralized (J, CPU)
1	276.2, 13.8 s	304.5, 2.2 s	363.1, 2.4 s
2	283.7, 14.1 s	306.7, 2.2 s	358.4, 2.5 s
3	283.4, 14.9 s	325.4, 2.5 s	332.6, 2.7 s
4	283.8, 12.2 s	317.7, 2.4 s	353.5, 2.4 s
5	273.2, 14.6 s	307.5, 2.7 s	334.9, 2.4 s
6	280.8, 14.4 s	312.7, 2.6 s	355.5, 2.7 s
7	264.9, 14.9 s	292.3, 2.4 s	346.5, 2.7 s
8	277.4, 14.1 s	306.5, 2.3 s	351.9, 2.4 s
9	279.6, 13.6 s	314.1, 2.8 s	351.3, 2.8 s
10	291.6, 15.9 s	324.7, 2.3 s	362.1, 2.6 s
Average	279.5, 14.2 s	311.2, 2.4 s	351.0, 2.6 s

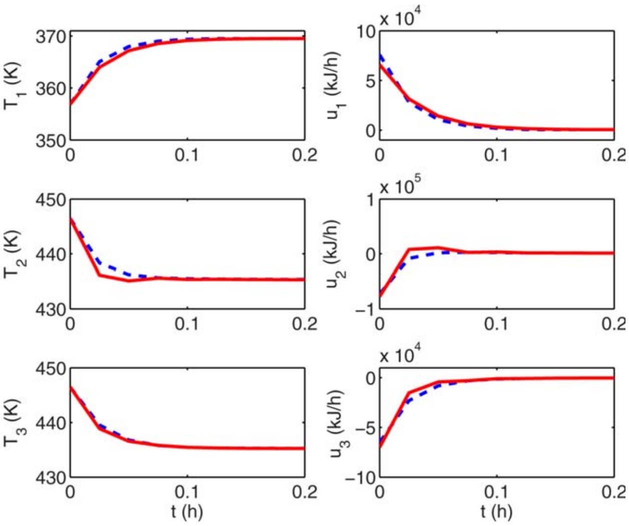

Figure 4. Temperature and input trajectories of closed-loop system under the proposed iterative distributed LMPC (dashed line) and the cooperative distributed LMPC (solid line).

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

In the second set of simulations, we compare the proposed iterative distributed LMPC implemented following Algorithm 2 with the iterative cooperative distributed LMPC. In the cooperative distributed LMPC, each local controller is designed based on the entire process model of Eq. 2 and optimizes the objective function of the centralized system, with each local controller communicating future input trajectories with all of the rest controllers at every iteration. The iteration is terminated if iteration times reaches $c_{\max}=10$ or if $|J^c(t_k)-J^{c-1}(t_k)| \leq 1$ where $J^c(t_k)$ denotes the objective function of the entire system evaluated at time instant t_k, iteration c.

We first compare the disturbance free case. The temperature and input trajectories of the nominal closed-loop system under the proposed iterative distributed LMPC and the cooperative distributed LMPC are shown in Figure 4. A comparison of the two iterative DMPC approaches is made in Table 3 with respect to the nominal performance index and total communication times. The results in Figure 4 and Table 3 show that the proposed iterative implementation obtains a comparable performance to that of the cooperative distributed LMPC with more than half communication times reduced. Note that the proposed iterative implementation improves the nominal control performance of the proposed noniterative algorithm.

Table 3. Nominal Performance Indices and Total Communication Times of the Proposed Iterative Distributed LMPC and the Cooperative Distributed LMPC

	Proposed Iterative Distributed LMPC	Cooperative Distributed LMPC
J	297.0	288.1
Communication times	69	162

Table 4. Performance Indices (*J*) and Communication Times (*CT*) of the Proposed Iterative Distributed LMPC and the Cooperative Distributed LMPC in 10 Runs

Run #	Proposed (J, CT)	Cooperative (J, CT)
1	290.4, 78	287.4, 144
2	294.2, 72	289.7, 156
3	313.9, 75	294.6, 156
4	306.5, 81	299.9, 180
5	296.0, 72	283.9, 156
6	299.4, 69	292.7, 144
7	280.6, 72	276.7, 162
8	295.6, 72	290.8, 156
9	300.8, 66	292.0, 162
10	313.6, 72	306.0, 156
Average	299.1, 73	291.3, 157

Finally, we compare the proposed iterative implementation with the iterative cooperative distributed LMPC with process disturbances. Ten runs with different sequences of disturbances for each of the two iterative control configurations are carried out. The performance indices and total communication times are shown in Table 4. These results agree with the nominal case as well.

Conclusions

In this work, we focused on distributed LMPC of nonlinear systems with neighbor-to-neighbor communication. A noniterative algorithm is first proposed, followed by an iterative implementation to improve the control performance. In both algorithms, only one-sided communication takes place from upstream subsystems to their downstream neighbors. In the distributed LMPC designs, each subsystem LMPC is based on an auxiliary Lyapunov-based controller that involves the subsystem state as well as the states of its upstream neighbors. Sufficient conditions for the closed-loop stability of the proposed algorithms were derived. Extensive simulations obtained from a reactor-separator process showed that the proposed distributed LMPC algorithms provide a balanced solution among existing LMPC implementations in terms of implementation complexity and achievable performance.

Acknowledgment

Financial support from Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

Literature Cited

1. Qin SJ, Badgwell TA. A survey of industrial model predictive control technology. *Control Eng Pract.* 2003;11:733–764.
2. Rawlings JB, Stewart BT. Coordinating multiple optimization-based controllers: new opportunities and challenges. *J. Process Control.* 2008;18:839–845.
3. Scattolini R. Architectures for distributed and hierarchical model predictive control: a review. *J. Process Control.* 2009;19:723–731.
4. Christofides PD, Scattolini R, Muñoz de la Peña D, Liu J. Distributed model predictive control: a tutorial review and future research directions, *Comput Chem Eng.* 2013;51:21–41.
5. 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.
6. Magni L, Scattolini R. Stabilizing decentralized model predictive control of nonlinear systems. *Automatica.* 2006;42:1231–1236.
7. Stanković SS, Stipanović DM, Šiljak DD. Decentralized dynamic output feedback for robust stabilization of a class of nonlinear interconnected systems. *Automatica.* 2007;43:861–867.
8. Baldea M, Daoutidis P, Kumar A. Dynamics and control of integrated networks with purge streams. *AIChE J.* 2006;52:1460–1472.
9. Jogwar SS, Torres AI, Daoutidis P. Networks with large solvent recycle: dynamics, hierarchical control, and a biorefinery application. *AIChE J.* 2012;58:1764–1777.
10. Baldea M, Touretzky CR. Nonlinear model predictive control of energy-integrated process systems. *Syst Control Lett.* 2013;62:723–731.
11. Baldea M, El-Farra NH, Ydstie BE. Dynamics and control of chemical process networks: integrating physics, communication and computation. *Comput Chem Eng.* 2013;51:42–54.
12. Cui H, Jacobsen EW. Performance limitations in decentralized control. *J Process Control.* 2002;12:485–494.
13. Davison EJ, Chang TN. Decentralized stabilization and pole assignment for general proper systems. *IEEE Trans Automat Contr.* 1990; 35:652–664.
14. Camponogara E, Jia D, Krogh BH, Talukdar S. Distributed model predictive control. *IEEE Control Syst Mag.* 2002;22:44–52.
15. Dunbar WB, Murray RM. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica.* 2006;42:549–558.
16. Jia D, Krogh BH. Distributed model predictive control. In: *Proceedings of the American Control Conference*. American Automatic Control Council, Arlington, VA, 2001;2767–2772.
17. Tippett MJ, Bao J. Distributed model predictive control based on dissipativity. *AIChE J.* 2013;59:787–804.
18. Al-Gherwi W, Budman H, Elkamel A. A robust distributed model predictive control algorithm. *J Process Control.* 2011;21:1127–1137.
19. 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*. Institute of Electrical and Electronics Engineers (IEEE), Seville, Spain, 2005;6680–6685.
20. Stewart BT, Venkat AN, Rawlings JB, Wright SJ, Pannocchia G. Coorperative distributed model predictive control. *Syst Contr Lett.* 2010;59:460–469.
21. Liu J, Muñoz de la Peña D, Christofides PD. Distributed model predictive control of nonlinear process systems. *AIChE J.* 2009;55: 1171–1184.
22. Liu J, Chen X, Muñoz de la Peña D, Christofides PD. Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. *AIChE J.* 2010;56:2137–2149.
23. Christofides PD, Liu J, Muñoz de la Peña D. Networked and distributed predictive control: methods and nonlinear process network applications. In: *Advances in Industrial Control Series*. London, England: Springer-Verlag, 2011.
24. Maestre JM, Muñoz de la Peña D, Camacho EF. Distributed model predictive control based on a cooperative game. *Opt Contr Appl Meth.* 2011;32:153–176.
25. Maestre JM, Muñoz de la Pena D, Camacho EF, Alamo T. Distributed model predictive control based on agent negotiation. *J Process Control.* 2011;21:685–697.
26. Cheng R, Forbes JF, Yip WS. Price-driven coordination method for solving plant-wide MPC problems. *J Process Control.* 2007;17:429–438.
27. Cheng R, Forbes JF, Yip WS. Dantzig-Wolfe decomposition and plant-wide MPC coordination. *Comput Chem Eng.* 2008;32:1507–1522.
28. Hassanzadeh B, Pakravesh H, Liu J, Forbes JF. Coordinated-distributed mpc of nonlinear systems based on price-driven coordination. In: *Proceedings of the American Control Conference*, American Automatic Control Council, Washington, DC, 2013; 3159–3164.
29. Scheu H, Marquardt W. Sensitivity-based coordination in distributed model predictive control. *J. Process Control.* 2011;21:715–728.
30. Sun Y, El-Farra NH. Quasi-decentralized model-based networked control of process systems. *Comput Chem Eng.* 2008;32:2016–2029.
31. Hu Y, El-Farra NH. Quasi-decentralized output feedback model predictive control of networked process systems with forecast-triggered communication. In: *Proceedings of the American Control Conference*. American Automatic Control Council, Washington, DC, 2013:2612–2617.
32. Farina M, Scattolini R. Distributed model predictive control: a non-cooperative algorithm with neighbor-to-neighbor communication for linear systems. *Automatica.* 2012;48:1088–1096.
33. Elliott MS, Rasmussen BP. Neighbor-communication model predictive control and HVAC systems. In: *Proceedings of the American*

Control Conference, American Automatic Control Council, Montreal, Canada, 2012:3020–3025.

34. Mhaskar P, El-Farra NH, Christofides PD. Predictive control of switched nonlinear systems with scheduled mode transitions. *IEEE Trans Automat Contr*. 2005;50:1670–1680.
35. Mahmood M, Mhaskar P. Lyapunov-based model predictive control of stochastic nonlinear systems. *Automatica*. 2012;48:2271–2276.
36. Khalil HK. Nonlinear systems. In: *Upper Saddle River*, 3rd ed., NJ: Prentice Hall, 2002.
37. Liu J, Ohran BJ, Muñoz de la Peña D, Christofides PD, Davis JF. Monitoring and handling of actuator faults in two-tier control systems for nonlinear processes. *Chem Eng Sci*. 2010;65:3179–3190.

Manuscript received June 20, 2014, and revision received July 25, 2014.