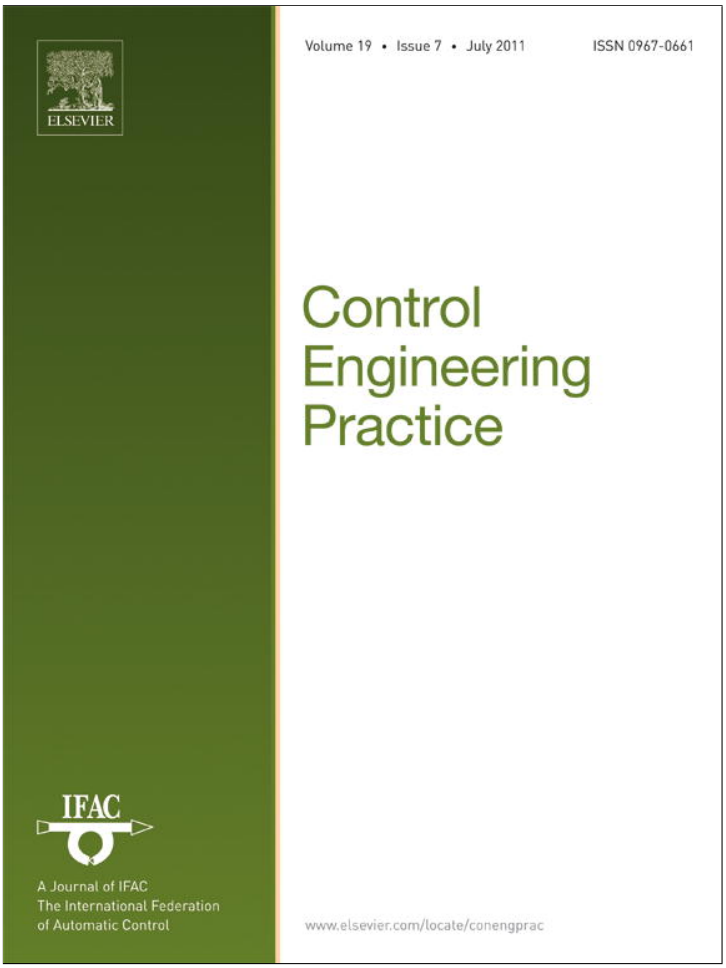
Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright

Author's personal copy

Control Engineering Practice 19 (2011) 757–769

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Distributed model predictive control over network information exchange for large-scale systems

Yi Zheng, Shaoyuan Li*, Ning Li

Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China

ARTICLE INFO

Article history:
Received 10 September 2009
Accepted 4 April 2011
Available online 4 May 2011

Keywords:
Distributed model predictive control
Model predictive control
Stability
Neighbourhood optimization

ABSTRACT

A class of large scale systems, which is naturally divided into many smaller interacting subsystems, are usually controlled by a distributed or decentralized control framework. In this paper, a novel distributed model predictive control (MPC) is proposed for improving the performance of entire system. In which each subsystem is controlled by a local MPC and these controllers exchange a reduced set of information with each other by network. The optimization index of each local MPC considers not only the performance of the corresponding subsystem but also that of its neighbours. The proposed architecture guarantees satisfactory performance under strong interactions among subsystems. A stability analysis is presented for the unconstrained distributed MPC and the provided stability results can be employed for tuning the controller. Experiment of the application to accelerated cooling process in a test rig is provided for validating the efficiency of the proposed method.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

There are class of complex large scale systems which are composed by many physically or geographically divided subsystems. Each subsystem interacts with some so called "neighbouring subsystems" by their states and their inputs. The technical target is to achieve some global performance of entire system (or a common goal of all subsystems).

The classical centralized control solution, where a control agent is able to acquire the information of the global system and could obtain a good global performance, is often impractical to apply to large scale system for some reasons: (1) there are hundreds of inputs and outputs. It requires a large computational efforts in online implementation; (2) when the centralized controller fails, the entire system is out of control and the control integrity cannot be guaranteed when a control component fails and (3) in some cases, e.g. in multi-intelligent vehicle system, the global information is unavailable to each controller.

The distributed (or decentralized) framework, where each subsystem is controlled by an independent controller, has the advantages of being flexible to system structure, error-tolerance, less computational efforts and no global information requirements (Du Xi, & Li, 2001; Vaccarini, Longhi, & Katebi, 2009). Thus the distributed control framework is usually adopted in this class of system (Du, Xi, & Li, 2001; Vaccarini, Longhi, & Katebi, 2009), in spite of that the dynamic performance of centralized framework is better than it. On the other hand, the development of DCS, fieldbus, communication network technologies in process industries allows the control technologies and methodologies to utilize their potentials for improving control. Thus, how to improve the global performance of each subsystem with the limited network communication or limited available information is a valuable problem.

Model predictive control (MPC), also called receding horizon control, is widely recognized as a high practical control technology with high performance. Where a control action sequence is obtained by solving, at each sampling instant, a finite horizon open-loop receding optimization problem and the first control action is applied to the processes (Maciejowski, 2002). It has been applied successfully to various linear (Li, Zheng, & Wang, 2008; Qin & Badgwell, 2003; Richalet, 1993), nonlinear (Peng, Nakano, & Shioya, 2007; Qin & Badgwell, 2000; Xu, Li, & Cai, 2005; Zheng, Li, & Wang, 2009) systems in the process industries and is becoming more widespread (Lee, Kumara, & Gautam, 2008; Qin & Badgwell, 2003; Zheng, Li, & Wang, 2011). Nowadays, the distributed framework of MPC, distributed MPC, is also gradually developing for the control of large scale systems.

Some distributed MPC formulations are available in the literatures (Camponogara, Jia, & Krogh, 2002; Dunbar, 2007; Dunbar & Murray, 2006; Du, Xi, & Li; Keviczky, Borrelli, & Balas, 2006; Lee, Kumara, & Gautam, 2008; Li, Zhang, & Zhu, 2005; Magni & Scattolini, 2006; Mercangoz & Iii, 2007; Richards & How, 2007; Vaccarini, Longhi, & Katebi, 2009; Venkat, Rawlings, & Wright, 2007; Venkat,

* Corresponding author. Tel./fax: +86 21 34204011.
E-mail address: syli@sjtu.edu.cn (S. Li).

0967-0661/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.conengprac.2011.04.003

Author's personal copy

Hiskens, & Rawlings, 2008; Zhang & Li, 2007; Zheng, Li, & Wang, 2009). Among them, the methods described in Dunbar and Murray (2006) and Keviczky, Borrelli, and Balas (2006) are proposed for a set of decoupled subsystems, and the extension of Dunbar and Murray (2006) could handle systems with weakly interacting subsystem dynamics (Dunbar, 2007). For large-scale linear time-invariant (LTI) systems, Camponogara, Jia, and Krogh (2002) present a distributed MPC scheme in which each local controller uses the state prediction of previous instant to approximate the state sequence of current instant in the procedure of computing an optimal solution. To improve the efficiency of distributed MPC solution, Li, Zhang, and Zhu (2005) developed an iterative algorithm for distributed MPC based on *Nash optimality*. The whole system will arrive at Nash equilibrium if the convergence condition of the algorithm is satisfied. Also, a distributed MPC method with guaranteed feasibility properties is presented in Venkat, Rawlings, and Wright (2007). However, as pointed out by the authors Li, Zhang, and Zhu (2005), Venkat, Hiskens, and Rawlings (2008), Venkat, Rawlings, and Wright (2007) and Zhang and Li (2007), the performance of the distributed MPC framework is, in most cases, worse than that of centralized MPC. In order to guarantee performance improvement and the appropriate communication burden among subsystems, an extended scheme based on a so called "*neighbourhood optimization*" is proposed in Zhang and Li (2007) for a class of serially connected processes, in which subsystems are interconnected by inputs. As for the class of system in which each subsystem interacts with some so called "neighbouring subsystems" by both their states and their inputs, Vaccarini, Longhi, and Katebi (2009) presents a decentralized MPC in which each local controller exchanges information with its neighbours to account for the interactions among subsystems. However, the optimization in each local controller is to pursue the performance of local subsystem. Camponogara, Jia, and Krogh (2002) presents a method to improve global optimality, while it is based on global information. Thus, how to improve the global performance using appropriate network resources is still a problem for this class of large scale system.

In this paper, a novel distributed MPC based on neighbourhood optimization for the large scale system mentioned above is developed, in which the optimization objective of each subsystem-based MPC considers not only the performance of local subsystem corresponding but also those of its neighbors. In the optimization, each local controller takes into account not only the impacts coming from its neighbors but also the impacts applied to its neighbors for improving global performance. The closed-loop stability analysis is also provided for guiding local MPCs tuning. Moreover, the performance of closed-loop system using proposed distributed MPC is analyzed and the application to accelerated cooling and controlled (ACC) process is presented to validate the efficiency of this method.

The contents are organized as follows. Section 2 describes the problem to be solved. Section 3 presents the proposed neighbourhood-optimization based distributed MPC, and gives its closed-form solution. Section 4 provides the stability condition of closed-loop system. Section 5 discusses the performance of proposed distributed MPC. The experiments of applying proposed distributed MPC to ACC test rig are presented in Section 6. Finally, a brief conclusion is drawn to summarize the study.

2. Problem description

2.1. System

For a class of large scale system with hundreds or thousands of inputs and outputs variables (e.g. Power and energy network, large chemical processes), since the centralized control is forbidden for the scale of system or the less flexibility when some errors occurred in one or several subsystems, the distributed framework is usually adopted in spite of the losing global performance. As shown in Fig. 1, the whole system is properly partitioned into several interconnected subsystems. Each subsystem is controlled by a local controller and these local controllers are interconnected by network. Although the problem of dividing system has been referred to in literatures, it is still not systematically solved (Bakule, 2008; Scattolini, 2009). In some cases, partitioning is natural in view of the process layout, see for example Rawlings and Stewart (2008) where chemical plants are considered. In other cases, the partitioning can be made by some mathematical methodologies (Niederlinski, 1971; Van Henten & Bontsema, 2009). In this paper, the system which has been natural partitioned in view of process layout is considered.

Without losing generality, suppose that the whole system is composed of n linear, discrete-time subsystems $\mathcal{S}_i, i=1,2,\ldots,n$, and each subsystem interacts with each other by both inputs and states, the state-space model of subsystem $\mathcal{S}_i$ can be expressed as

$$\begin{aligned}\boldsymbol{x}_i(k+1) &= \boldsymbol{A}_{ii}\boldsymbol{x}_i(k)+\boldsymbol{B}_{ii}u_i(k)+\sum_{j=1(j\neq i)}^{n}\boldsymbol{A}_{ij}\boldsymbol{x}_j(k)+\sum_{j=1(j\neq i)}^{n}\boldsymbol{B}_{ij}u_j(k)\\ \boldsymbol{y}_i(k) &= \boldsymbol{C}_{ii}\boldsymbol{x}_i(k)+\sum_{j=1(j\neq i)}^{n}\boldsymbol{C}_{ij}\boldsymbol{x}_j(k)\end{aligned} \tag{1}$$

where vectors $\boldsymbol{x}_i\in\mathbb{R}^{n_{x_i}}, \boldsymbol{u}_i\in\mathbb{R}^{n_{u_i}}$ and $\boldsymbol{y}_i\in\mathbb{R}^{n_{y_i}}$ are the local state, control input and output vectors, respectively. When at least one of matrices $\boldsymbol{A}_{ij}, \boldsymbol{B}_{ij}, \boldsymbol{C}_{ij}$ is not null, it is said that $\mathcal{S}_j$ interacts with $\mathcal{S}_i$. The whole system can be expressed as

$$\begin{aligned}\boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k)+\boldsymbol{B}u(k)\\ \boldsymbol{y}(k) &= \boldsymbol{C}\boldsymbol{x}(k)\end{aligned} \tag{2}$$

where $\boldsymbol{x}\in\mathbb{R}^{n_x}, \boldsymbol{u}\in\mathbb{R}^{n_u}$ and $\boldsymbol{y}\in\mathbb{R}^{n_y}$, are state, control input and output vectors, respectively. The control objective of this system is minimizing a global performance index $J(k)$ at time k, and

$$J(k)=\sum_{i=1}^{n}\left[\sum_{l=1}^{P}\|\boldsymbol{y}_i(k+l)-\boldsymbol{y}_i^d(k+l)\|_{\boldsymbol{Q}_i}^2+\sum_{l=1}^{M}\|\Delta\boldsymbol{u}_i(k+l-1)\|_{\boldsymbol{R}_i}^2\right] \tag{3}$$

where $\boldsymbol{Q}_i$ and $\boldsymbol{R}_i$ are weight matrices, $P,M\in\mathbb{N}$ are predictive horizon and control horizon, respectively, and $P\geq M$, $\boldsymbol{y}_i^d$ is the set-point of subsystem $\mathcal{S}_i$, $\Delta\boldsymbol{u}_i(k)=\boldsymbol{u}_i(k)-\Delta\boldsymbol{u}_i(k-1)$ is the input increment vector of subsystem $\mathcal{S}_i$.

Moreover, in many situations, the communication resources are not unlimited for the safety reason and communication bandwidth limitation, or the global information is unavailable to every subsystem due to the physical or man-made reasons. Those require a simple structure of local controller. Thus, as pointed out in Scattolini (2009), how to improve the performance of entire

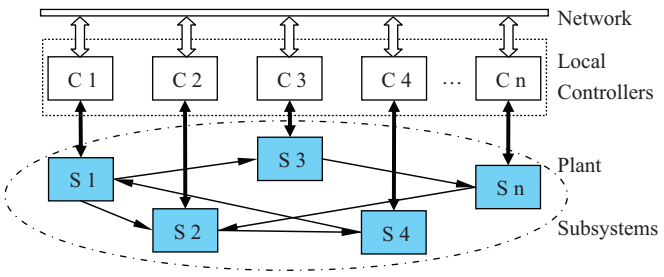


Fig. 1. The structure of distributed system and distributed control framework.

Author's personal copy

system is still a challenge for this class of system under the distributed control framework with simply control structure.

2.2. Existing methods

There are two classes of distributed MPCs appeared in literatures for the large scale systems described above.

(A) Distributed algorithms where each local controller minimize following local optimization objective:

$$J_i(k)=\sum_{l=1}^{P}\|\boldsymbol{y}_i(k+l)-\boldsymbol{y}_i^d(k+l)\|_{\boldsymbol{Q}_i}^2+\sum_{l=1}^{M}\|\Delta\boldsymbol{u}_i(k+l-1)\|_{\boldsymbol{R}_i}^2$$

When computing optimal solution, each local controller exchanges estimation states with its neighbours, therefore improves the performance of closed-loop subsystem. However the performance of other subsystems is not considered in this optimization.

(B) Distributed algorithms where each local controller minimizes a global cost function

$$J(k)=\sum_{i=1}^{n}J_i(k)$$

This strategy could achieve a good performance closing to the centralized MPC. However, this strategy requires much more communication resources and the structure of controller is relatively complex.

In this paper a method based on neighbourhood optimization is proposed for the large scale system in which each subsystem interacts with each other by both inputs and states. The goal of it is to achieve a significantly improving performance of entire system with little increasing of the network resources.

3. Neighbourhood-optimization distributed MPC (ND-MPC)

The proposed control architecture is based on a set of independent MPC controllers $\mathcal{C}_i$, $i=1,2,\ldots,n$, for each subsystem $\mathcal{S}_i$. Each controller could exchange information with its neighbours through network. To discuss the control methodology proposed in this paper, the simplifying hypothesis of accessible local states $x_i(k)$ and Assumption 1 are considered. Moreover, Definition 1 and notations listed in Table 1 are defined to describe the proposed methodology clearly.

Assumption 1.

(a) Controllers are synchronous;
(b) Controllers communicate only once within a sampling time interval;
(c) Communication cannel introduces a delay of a single sampling time interval.

This set of assumptions is not restrictive. The controllers are synchronous and are not so strong because the sampling interval is usually rather long compared to the computational time in process control. The assumption (b) of single information exchange with a sampling time interval is due to the necessity of minimizing the amount of data exchange through the network. In real situations an instantaneous data transfer is not possible; therefore assumption (c) of unit delay is required.

Definitions 1. ***Neighbouring subsystem***: subsystem $\mathcal{S}_i$ interacts with $\mathcal{S}_j$, and the states, outputs of subsystem $\mathcal{S}_i$ are affected by subsystem $\mathcal{S}_j$. In this case $\mathcal{S}_j$ is called input neighbouring subsystem of $\mathcal{S}_i$ and $\mathcal{S}_i$ is called the output neighbouring subsystem of $\mathcal{S}_i$. $\mathcal{S}_i$ and $\mathcal{S}_j$ are said neighbouring subsystems or neighbours.

Table 1
Notations used in this paper.

Notations	Explanations
$\hat{\boldsymbol{x}}_i(l\|h)$, $\hat{\boldsymbol{y}}_i(l\|h)$,	The predictions of $\boldsymbol{x}_i(l)$ and $\boldsymbol{y}_i(l)$ computed at time h, and $l,h\in\mathbb{N}$, $h<l$;
$\Delta\boldsymbol{u}_i(l\|h)$, $\boldsymbol{u}_i(l\|h)$	The input $\boldsymbol{u}_i(l)$ and the input increment $\Delta\boldsymbol{u}_i(l)$ computed by controller $\mathcal{C}_i$ at time h, $l,h\in\mathbb{N}$ and $h<l$;
$\boldsymbol{y}_i^d(l\|h)$	The set-point of $\boldsymbol{y}_i(l\|h)$;
$\widehat{\boldsymbol{x}}_i(k)$, $\widehat{\boldsymbol{y}}_i(k)$,	The state and output vectors of the output neighbourhood of $\mathcal{S}_i$. $\widehat{\boldsymbol{x}}_i(k)=[\boldsymbol{x}_i^T(k)\ \ \boldsymbol{x}_{i_1}^T(k)\ \ \cdots\ \ \boldsymbol{x}_{i_m}^T(k)]^T$ and $\widehat{\boldsymbol{y}}_i(k)=[\boldsymbol{y}_i^T(k)\ \ \boldsymbol{y}_{i_1}^T(k)\ \ \cdots\ \ \boldsymbol{y}_{i_m}^T(k)]^T$, m is the number of the output-neighbours of $\mathcal{S}_i$;
$\widehat{\boldsymbol{w}}_i(k)$, $\widehat{\boldsymbol{v}}_i(k)$	The interactions act on the state and output of output-neighbourhood of $\mathcal{S}_i$. See Eqs. (9) and (10);
$\widehat{\hat{\boldsymbol{x}}}_i(l\|h)$, $\widehat{\hat{\boldsymbol{y}}}_i(l\|h)$	The predictions of $\widehat{\boldsymbol{x}}_i(l)$ and $\widehat{\boldsymbol{y}}_i(l)$ computed at time h, $l,h\in\mathbb{N}$ and $h<l$;
$\widehat{\hat{\boldsymbol{w}}}_i(l\|h)$, $\widehat{\hat{\boldsymbol{v}}}_i(l\|h)$	The estimations of $\widehat{\boldsymbol{w}}_i(l)$ and $\widehat{\boldsymbol{v}}_i(l)$ computed at time h, $l,h\in\mathbb{N}$ and $h<l$;
$\widehat{\boldsymbol{y}}_i^d(l\|h)$	The set-point of $\widehat{\boldsymbol{y}}_i(l\|h)$;
$\boldsymbol{U}_i(l,p\|h)$	A complete input vector, $\boldsymbol{U}_i(l,p\|h)=[\boldsymbol{u}_i^T(l\|h)\ \ \boldsymbol{u}_i^T(l+1\|h)\ \ \cdots\ \ \boldsymbol{u}_i^T(l+p\|h)]^T$, p, $l,h\in\mathbb{N}$ and $h<l$;
$\Delta\boldsymbol{U}_i(l,p\|h)$	Input increment sequence vector, $\Delta\boldsymbol{U}_i(l,p\|h)=[\Delta\boldsymbol{u}_i^T(l\|h)\ \ \Delta\boldsymbol{u}_i^T(l+1\|h)\ \ \cdots\ \ \Delta\boldsymbol{u}_i^T(l+p\|h)]^T$, $h<l$;
$\boldsymbol{U}(l,p\|h)$	A complete stacked input vector, $\boldsymbol{U}(l,p\|h)=[\boldsymbol{u}_1^T(l\|h)\ \ \cdots\ \ \boldsymbol{u}_n^T(l\|h)\ \ \cdots\ \ \boldsymbol{u}_1^T(l+p\|h)\ \ \cdots\ \ \boldsymbol{u}_n^T(l+p\|h)]^T$;
$\hat{\boldsymbol{X}}_i(l,p\|h)$	A stacked distributed state vector, $\hat{\boldsymbol{X}}_i(l,p\|h)=[\hat{\boldsymbol{x}}_i^T(l\|h)\ \ \hat{\boldsymbol{x}}_i^T(l+1\|h)\ \ \cdots\ \ \hat{\boldsymbol{x}}_i^T(l+p\|h)]^T$;
$\hat{\boldsymbol{X}}(l,p\|h)$	A complete stacked state vector, $\hat{\boldsymbol{X}}(l,p\|h)=[\hat{\boldsymbol{x}}_1^T(l\|h)\ \ \cdots\ \ \hat{\boldsymbol{x}}_n^T(l\|h)\ \ \cdots\ \ \hat{\boldsymbol{x}}_1^T(l+p\|h)\ \ \cdots\ \ \hat{\boldsymbol{x}}_n^T(l+p\|h)]^T$;
$\hat{\hat{\boldsymbol{X}}}_i(l,p\|h)$	A stacked state vector, $\hat{\hat{\boldsymbol{X}}}_i(l,p\|h)=[\hat{\hat{\boldsymbol{x}}}_i^T(l\|h)\hat{\hat{\boldsymbol{x}}}_i^T(l+1\|h)\cdots\hat{\hat{\boldsymbol{x}}}_i^T(l+p\|h)]^T$, p, $l,h\in\mathbb{N}$ and $h<l$;
$\hat{\hat{\boldsymbol{Y}}}_i(l,p\|h)$	A stacked output vector, $\hat{\hat{\boldsymbol{Y}}}_i(l,p\|h)=[\hat{\hat{\boldsymbol{y}}}_i^T(l\|h)\hat{\hat{\boldsymbol{y}}}_i^T(l+1\|h)\cdots\hat{\hat{\boldsymbol{y}}}_i^T(l+p\|h)]^T$, p, $l,h\in\mathbb{N}$ and $h<l$;
$\hat{\hat{\boldsymbol{Y}}}_i^d(l,p\|h)$	The set-point of $\hat{\hat{\boldsymbol{Y}}}_i(l,p\|h)$;
$\hat{\hat{\boldsymbol{W}}}_i(l,p\|h)$	A stacked interaction vector, $[\hat{\hat{\boldsymbol{w}}}_i^T(l\|h)\hat{\hat{\boldsymbol{w}}}_i^T(l+1\|h)\cdots\hat{\hat{\boldsymbol{w}}}_i^T(l+p\|h)]^T$, p, $l,h\in\mathbb{N}$ and $h<l$;
$\hat{\hat{\boldsymbol{V}}}_i(l,P\|h)$	A stacked interaction vector, $[\hat{\hat{\boldsymbol{v}}}_i^T(l\|h)\hat{\hat{\boldsymbol{v}}}_i^T(l+1\|h)\cdots\hat{\hat{\boldsymbol{v}}}_i^T(l+p\|h)]^T$, p, $l,h\in\mathbb{N}$ and $h<l$;
$\hat{\mathbb{X}}=(l,p\|h)$	A complete stacked state vector, $\hat{\mathbb{X}}=(l,p\|h)=[\hat{\boldsymbol{X}}_1^T(l,p\|h)\ \ \cdots\ \ \hat{\boldsymbol{X}}_m^T(l,p\|h)]^T$;
$\mathbb{U}=(l,p\|h)$	A complete stacked state vector, $\mathbb{U}=(l,p\|h)=[\boldsymbol{U}_1^T(l,p\|h)\ \ \cdots\ \ \boldsymbol{U}=\ {}_m^T(l,p\|h)]^T$.

Author's personal copy

***Neighbourhood of a subsystem*:** the input (output)-neighbourhood $\mathcal{N}_i^{\text{in}}(\mathcal{N}_i^{\text{out}})$ of subsystem $\mathcal{S}_i$ is the set of all its input (output)-neighbours

$$\mathcal{N}_i^{\text{in}} = \{\mathcal{S}_i, \mathcal{S}_j | \mathcal{S}_j \text{ is an input neighbour of } \mathcal{S}_i\}$$

$$\mathcal{N}_i^{\text{out}} = \{\mathcal{S}_i, \mathcal{S}_j | \mathcal{S}_j \text{ is an output neighbour of } \mathcal{S}_i\}$$

The neighbourhood $\mathcal{N}_i$ of subsystem $\mathcal{S}_i$ is the set of all its neighbours

$$\mathcal{N}_i = \mathcal{N}_i^{\text{in}} \cup \mathcal{N}_i^{\text{out}}.$$

3.1. ND-MPC formulation

3.1.1. Performance index

For the large scale system considered in this paper, the global performance index (3) can be decomposed in terms of the local index J_i for each subsystem $\mathcal{S}_i$, $i = 1,2,\ldots,n$ (Katebi & Johnson, 1997)

$$J_i(k) = \sum_{l=1}^{P} \|\hat{\boldsymbol{y}}_i(k+l|k) - \boldsymbol{y}_i^d(k+l|k)\|_{\boldsymbol{Q}_i}^2 + \sum_{l=1}^{M} \|\Delta \boldsymbol{u}_i(k+l-1|k)\|_{\boldsymbol{R}_i}^2 \tag{4}$$

The local control decision of $\mathcal{S}_i$ is computed by solving the optimization problem $\min_{\Delta \boldsymbol{U}(k,M|k)} J_i(k)$ with local input/output variables and constraints in the distributed MPC based on the state (or input) estimations of neighbors at time $k-1$ (Vaccarini, Longhi, & Katebi, 2009) or Nash optimality (Li, Zhang, & Zhu, 2005). The method present in Vaccarini, Longhi, and Katebi (2009) adopts this local index.

However, since the state evolution of output-neighbours of subsystem $\mathcal{S}_i$ is affected by the control decision of subsystem $\mathcal{S}_i$, see Eq. (1), the performance of these neighbours may be destroyed by improper control decision of $\mathcal{S}_i$ in some cases. To solve this problem, the so called "*Neighbourhood optimization*" (Zhang & Li, 2007; Zheng, Li, & Wang, 2009) is adopted and the performance index is expressed as

$$\begin{aligned}\bar{J}_i(k) &= \sum_{j \in \mathcal{N}_i^{\text{out}}} J_i(k) = \sum_{j \in \mathcal{N}_i^{\text{out}}} \left[\sum_{l=1}^{P} \|\hat{\boldsymbol{y}}_j(k+l|k) - \boldsymbol{y}_j^d(k+l|k)\|_{\boldsymbol{Q}_j}^2 \right. \\ &\quad \left. + \sum_{l=1}^{M} \|\Delta \boldsymbol{u}_j(k+l-1|k)\|_{\boldsymbol{R}_j}^2 \right]\end{aligned} \tag{5}$$

Since $\Delta \boldsymbol{u}_j(k+l-1|k)$ $(j \in \mathcal{N}_i^{\text{out}}, j \neq i, l = 1,\ldots,M)$ is unknown and independent to the control decision of $\mathcal{S}_i$, $\Delta \boldsymbol{u}_j(k+l-1|k-1)$ is used to approximate $\Delta \boldsymbol{u}_j(k+l-1|k)$. Then, Eq. (5) becomes

$$\begin{aligned}\bar{J}_i(k) &= \sum_{j \in \mathcal{N}_i^{\text{out}}} \sum_{l=1}^{P} \|\hat{\boldsymbol{y}}_j(k+l|k) - \boldsymbol{y}_j^d(k+l|k)\|_{\boldsymbol{Q}_j}^2 + \sum_{l=1}^{M} \|\Delta \boldsymbol{u}_i(k+l-1|k)\|_{\boldsymbol{R}_i}^2 \\ &\quad + \sum_{j \in \mathcal{N}_i^{\text{out}}, j \neq i} \sum_{l=1}^{M} \|\Delta \boldsymbol{u}_i(k+l-1|k-1)\|_{\boldsymbol{R}_j}^2 \\ &= \sum_{j \in \mathcal{N}_i^{\text{out}}} \sum_{l=1}^{P} \|\hat{\boldsymbol{y}}_j(k+l|k) - \boldsymbol{y}_j^d(k+l|k)\|_{\boldsymbol{Q}_j}^2 \\ &\quad + \sum_{l=1}^{M} \|\Delta \boldsymbol{u}_i(k+l-1|k)\|_{\boldsymbol{R}_i}^2 + \text{Constant}\end{aligned}$$

For simplifying reason, redefine $\bar{J}_i(k)$ as

$$\bar{J}_i(k) = \sum_{l=1}^{P} \|\hat{\boldsymbol{y}}_i(k+l|k) - \boldsymbol{y}_i^d(k+l|k)\|_{\widehat{\boldsymbol{Q}}_i}^2 + \sum_{l=1}^{M} \|\Delta \boldsymbol{u}_i(k+l-1|k)\|_{\boldsymbol{R}_i}^2 \tag{6}$$

where $\widehat{\boldsymbol{Q}}_i = diag(\boldsymbol{Q}_i, \boldsymbol{Q}_{i_1}, \ldots, \boldsymbol{Q}_{i_b})$.

The optimization index $\bar{J}_i(k)$ considers not only the performance of subsystem $\mathcal{S}_i$ but also that of the output-neighbours of $\mathcal{S}_i$. The impacts of the control decision of $\mathcal{S}_i$ to $\mathcal{S}_j \in \mathcal{N}_i^{\text{out}}$ are fully considered in the neighbourhood optimization, and therefore the global performance improving is guaranteed. It should be noticed that the global performance may be farther improved if using the optimization objective (3) in each subsystem, but it requires a high quality and complicated network communication and introduces more complex computation.

3.1.2. Prediction model

Since the state evolution of $\mathcal{S}_j \in \mathcal{N}_i^{\text{out}}$ is affected by $\boldsymbol{u}_i(k)$, to improve the predictive precision, subsystem $\mathcal{S}_i$ and its output-neighbours should be considered as one relatively large integral subsystem when predicting the future states of $\mathcal{S}_i$ and its output-neighbours. Assume that the number of output neighbours of $\mathcal{S}_i$ is m, then the state evolution model of the output-neighbourhood of $\mathcal{S}_i$ can be easily deduced by Eq. (2) and expressed as

$$\begin{cases} \widehat{\boldsymbol{x}}_i(k+1) = \widehat{\boldsymbol{A}}_i \widehat{\boldsymbol{x}}_i(k) + \widehat{\boldsymbol{B}}_i \boldsymbol{u}_i(k) + \widehat{\boldsymbol{w}}_i(k) \\ \widehat{\boldsymbol{y}}_i(k) = \widehat{\boldsymbol{C}}_i \widehat{\boldsymbol{x}}_i(k) + \widehat{\boldsymbol{v}}_i(k) \end{cases} \tag{7}$$

where

$$\widehat{\boldsymbol{A}}_i \quad \left[\widehat{\boldsymbol{A}}_i^{(1)} \quad \widehat{\boldsymbol{A}}_i^{(2)}\right] \quad \left[\begin{array}{c|ccc} \boldsymbol{A}_{ii} & \boldsymbol{A}_{ii_1} & \cdots & \boldsymbol{A}_{ii_m} \\ \hline \boldsymbol{A}_{i_1 i} & \boldsymbol{A}_{i_1 i_1} & \cdots & \boldsymbol{A}_{i_1 i_m} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{i_m i} & \boldsymbol{A}_{i_m i_1} & \cdots & \boldsymbol{A}_{i_m i_m} \end{array}\right], \quad \widehat{\boldsymbol{B}}_i = \begin{bmatrix} \boldsymbol{B}_{ii} \\ \boldsymbol{B}_{i_1 i} \\ \vdots \\ \boldsymbol{B}_{i_m i} \end{bmatrix},$$

$$\widehat{\boldsymbol{C}}_i = \begin{bmatrix} \boldsymbol{C}_{ii} & \boldsymbol{C}_{ii_1} & \cdots & \boldsymbol{C}_{ii_m} \\ \boldsymbol{C}_{i_1 i} & \boldsymbol{C}_{i_1 i_1} & \cdots & \boldsymbol{C}_{i_1 i_m} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{C}_{i_m i} & \boldsymbol{C}_{i_m i_1} & \cdots & \boldsymbol{C}_{i_m i_m} \end{bmatrix} \tag{8}$$

$$\widehat{\boldsymbol{w}}_i(k) = \begin{bmatrix} \sum_{j \in \mathcal{N}_i^{\text{in}}, j \neq i} \boldsymbol{B}_{ij} u_j(k) + \boldsymbol{0} \\ \sum_{j \in \mathcal{N}_{i_1}^{\text{in}}, j \neq i} \boldsymbol{B}_{i_1 j} u_j(k) + \sum_{j \in \mathcal{N}_{i_1}^{\text{in}}, j \notin \mathcal{N}_i^{\text{out}}} \boldsymbol{A}_{i_1 j} x_j(k) \\ \vdots \\ \sum_{j \in \mathcal{N}_{i_m}^{\text{in}}, j \neq i} \boldsymbol{B}_{i_m j} u_j(k) + \sum_{j \in \mathcal{N}_{i_m}^{\text{in}}, j \notin \mathcal{N}_i^{\text{out}}} \boldsymbol{A}_{i_m j} x_j(k) \end{bmatrix} \tag{9}$$

$$\widehat{\boldsymbol{v}}_i(k) = \begin{bmatrix} \boldsymbol{0} \\ \sum_{j \in \mathcal{N}_{i_1}^{\text{in}}, j \notin \mathcal{N}_i^{\text{out}}} \boldsymbol{C}_{i_1 j} x_j(k) \\ \vdots \\ \sum_{j \in \mathcal{N}_{i_m}^{\text{in}}, j \notin \mathcal{N}_i^{\text{out}}} \boldsymbol{C}_{i_m j} x_j(k) \end{bmatrix} \tag{10}$$

It should be noticed that the input of this neighbourhood model is still the input of $\mathcal{S}_i$, and the inputs of $\mathcal{S}_j \in \mathcal{N}_i^{\text{out}}, j \neq i$ are regarded as disturbances. It is because that each local MPC can only determine the manipulated variables of the corresponding subsystem.

Due to the unit delay introduced by the network (see Assumption 1), the information of other subsystems is available only after one sampling time interval. Therefore controller $\mathcal{C}_i$ uses $\widehat{\boldsymbol{w}}_i(k+l-s|k-1))$ and $\widehat{\boldsymbol{v}}_i(k+l|k-1)$ computed on the basis of information related to time $k-1$ to estimate the interactions, and the initial states of output-neighbours are substituted with $\hat{\boldsymbol{x}}_{i_h}^T(k|k-1)$ $(h = 1,\ldots,m)$. For $i = 1,\ldots,n$, define

$$\widehat{\boldsymbol{x}}_i(k|k) = \left[\boldsymbol{x}_i^T(k|k) \quad \hat{\boldsymbol{x}}_{i_1}^T(k|k-1) \quad \cdots \quad \hat{\boldsymbol{x}}_{i_m}^T(k|k-1)\right]^T \tag{11}$$

Then the states and outputs of the output-neighbourhood in l-step ahead can be predicted by

$$\begin{cases} \widehat{\hat{\boldsymbol{x}}}_i(k+l|k) = \widehat{\boldsymbol{A}}_i^l \widehat{\boldsymbol{x}}_i(k|k) + \sum_{s=1}^{l} \widehat{\boldsymbol{A}}_i^{s-1} \widehat{\boldsymbol{B}}_i \boldsymbol{u}_i(k+l-s|k) + \sum_{s=1}^{l} \widehat{\boldsymbol{A}}_i^{s-1} \widehat{\boldsymbol{w}}_i(k+l-s|k-1)) \\ \hat{\boldsymbol{y}}_i(k+l|k) = \widehat{\boldsymbol{C}}_i \widehat{\hat{\boldsymbol{x}}}_i(k+l|k) + \hat{\boldsymbol{v}}_i(k+l|k-1) \end{cases} \tag{12}$$

Author's personal copy

3.1.3. Optimization problem

For each independent controller $\mathcal{C}_i$, $i=1,\ldots,n$, the unconstrained ND-MPC problem with prediction horizon P and control horizon M, $M<P$, at time k becomes to solve following optimization problem

$$\min_{\Delta \boldsymbol{U}_i(k,M|k)} \bar{J}_i(k) = \sum_{l=1}^{P} \|\hat{\boldsymbol{y}}_i(k+l|k) - \boldsymbol{y}_i^d(k+l|k)\|_{\bar{\boldsymbol{Q}}_i}^2 + \sum_{l=1}^{M} \|\Delta \boldsymbol{u}_i(k+l-1|k)\|_{\bar{\boldsymbol{R}}_i}^2 \quad \text{s.t. Eq. (12)} \tag{13}$$

At time k, the interaction predictions $\hat{\widehat{\boldsymbol{w}}}_i(k+l-1|k-1)$ and $\hat{\widehat{\boldsymbol{v}}}_i(k+l|k-1)$, together with $\widehat{\boldsymbol{x}}_i(k|k)$ are used to resolve the optimization problem (13) in each $\mathcal{C}_i$, $(i=1,\ldots,n)$. The first element of the optimal solution $\Delta \boldsymbol{U}_i^*(k)$ is selected and $\boldsymbol{u}_i(k)=\boldsymbol{u}_i(k-1)+\Delta \boldsymbol{u}(k|k)$ is applied to $\mathcal{S}_i$. Then, by Eq. (12), each local controller estimates the future state trajectory over the prediction horizon and broadcasts it in network together with the optimal control sequence over the control horizon. At time $k+1$, each local controller uses this information for evaluating the interaction predictions and the whole procedure is repeated.

The only information that each $\mathcal{C}_i$, $i=1,\ldots,n$, needs is the future behaviour of $\mathcal{S}_j \in \mathcal{N}_i$ and $\mathcal{S}_g \in \mathcal{N}_j$. Similarly, $\mathcal{C}_i$ broadcasts the future behaviour of $\mathcal{S}_i$ to the controller of $\mathcal{S}_j \in \mathcal{N}_i$ and controller of $\mathcal{S}_g \in \mathcal{N}_j$.

3.2. Closed-form solution

The main result of this subsection is the computation of the closed-form solution to the ND-MPC proposed. For this purpose, expressions of the interaction prediction and the state prediction are provided first. Define that

where $\boldsymbol{A}_{i,j}$, $\boldsymbol{B}_{i,j}$and $\boldsymbol{C}_{i,j}$ are zero blocks of congruent dimensions if $\mathcal{S}_j \notin \mathcal{N}_h^{\text{in}}$ ($\mathcal{S}_h \in \mathcal{N}_i^{\text{out}}$). Moreover define

$$n_u = \sum_{l=1}^{n} n_{u_l}, \quad \tilde{\boldsymbol{\Gamma}} = \begin{bmatrix} \boldsymbol{0}_{(M-1)n_u \times n_u} & \boldsymbol{I}_{(M-1)n_u} \\ \boldsymbol{0}_{n_u \times (M-1)n_u} & \boldsymbol{I}_{n_u} \\ \vdots & \vdots \\ \boldsymbol{0}_{n_u \times (M-1)n_u} & \boldsymbol{I}_{n_u} \end{bmatrix}, \quad \tilde{\boldsymbol{B}}_i = \tilde{\bar{\boldsymbol{B}}}_i \tilde{\boldsymbol{\Gamma}} \tag{18}$$

Then, following Lemmas can be deduced based on definitions (14)–(18). Proofs of the lemmas can be found in Appendixes.

Lemma 1 (Interaction prediction). *Under **Assumptions 1**, for each controller $\mathcal{C}_i$, $i=1,\ldots,n$, the stacked predictions of the interaction vectors at time k, based on the information computed at time $k-1$, are given by*

$$\begin{aligned} \hat{\widehat{\boldsymbol{W}}}_i(k,P|k-1) &= \tilde{\boldsymbol{A}}_{i1}\hat{\boldsymbol{X}}(k,P|k-1) + \tilde{\boldsymbol{B}}_i \boldsymbol{U}(k-1,M|k-1), \\ \hat{\widehat{\boldsymbol{V}}}_i(k,P|k-1) &= \tilde{\boldsymbol{C}}_i \hat{\boldsymbol{X}}(k,P|k-1). \end{aligned} \tag{19}$$

Lemma 2 (State prediction). *Under **Assumptions 1**, for each controller $\mathcal{C}_i$, $i=1,\ldots,n$, the stacked predictions of state and output of the output-neighbourhood of subsystem $\mathcal{S}_i$ at time k are expressed by*

$$\begin{cases} \hat{\widehat{\boldsymbol{X}}}_i(k+1,P|k) = \overline{\boldsymbol{S}}_i[\overline{\boldsymbol{A}}_i^{(1)}\hat{x}(k|k) + \overline{\boldsymbol{B}}_i \boldsymbol{U}_i(k,M|k) + \tilde{\boldsymbol{A}}_i \hat{\boldsymbol{X}}(k,P|k-1) \\ \qquad\qquad + \tilde{\boldsymbol{B}}_i \boldsymbol{U}(k-1,M|k-1)], \\ \hat{\widehat{\boldsymbol{Y}}}_i(k+1,P|k) = \overline{\boldsymbol{C}}_i \hat{\widehat{\boldsymbol{X}}}_i(k+1,P|k) + \boldsymbol{T}_i \tilde{\boldsymbol{C}}_i \hat{\boldsymbol{X}}(k+1,P|k-1). \end{cases} \tag{20}$$

where

$$\overline{\boldsymbol{A}}_i = \begin{bmatrix} \overline{\boldsymbol{A}}_i^{(1)} & \overline{\boldsymbol{A}}_i^{(2)} \end{bmatrix} = \begin{bmatrix} \widehat{\boldsymbol{A}}_i^{(1)} & \widehat{\boldsymbol{A}}_i^{(2)} \\ \boldsymbol{0}_{Pn_{\widehat{\mathcal{Y}}_i} \times n_{\mathcal{X}_i}} & \boldsymbol{0}_{Pn_{\widehat{\mathcal{Y}}_i} \times (n_{\widehat{\mathcal{X}}_i} - n_{\mathcal{X}_i})} \end{bmatrix},$$

$$\tilde{\boldsymbol{A}}_i(1) = diag_P\left\{ \underbrace{\begin{bmatrix} \boldsymbol{A}_{i,1} & \cdots & \boldsymbol{A}_{i,i-1} & \boldsymbol{0}_{n_{\mathcal{X}_i} \times n_{\mathcal{X}_i}} & \boldsymbol{A}_{i,i+1} & \cdots & \boldsymbol{A}_{i,i_1-1} & \boldsymbol{0}_{n_{\mathcal{X}_i} \times n_{\mathcal{X}_{i1}}} & \boldsymbol{A}_{i,i_1+1} & \cdots & \boldsymbol{A}_{i,i_m-1} & \boldsymbol{0}_{n_{\mathcal{X}_i} \times n_{\mathcal{X}_{im}}} & \boldsymbol{A}_{i,i_m+1} & \cdots & \boldsymbol{A}_{i,n} \\ \boldsymbol{A}_{i_1,1} & \cdots & \boldsymbol{A}_{i_1,i-1} & \boldsymbol{0}_{n_{\mathcal{X}_{i1}} \times n_{\mathcal{X}_i}} & \boldsymbol{A}_{i_1,i+1} & \cdots & \boldsymbol{A}_{i_1,i_1-1} & \boldsymbol{0}_{n_{\mathcal{X}_{i1}} \times n_{\mathcal{X}_{i1}}} & \boldsymbol{A}_{i_1,i_1+1} & \cdots & \boldsymbol{A}_{i_1,i_m-1} & \boldsymbol{0}_{n_{\mathcal{X}_{i1}} \times n_{\mathcal{X}_{im}}} & \boldsymbol{A}_{i_1,i_m+1} & \cdots & \boldsymbol{A}_{i_1,n} \\ \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ \boldsymbol{A}_{i_m,1} & \cdots & \boldsymbol{A}_{i_m,i-1} & \boldsymbol{0}_{n_{\mathcal{X}_{im}} \times n_{\mathcal{X}_i}} & \boldsymbol{A}_{i_m,i+1} & \cdots & \boldsymbol{A}_{i_m,i_1-1} & \boldsymbol{0}_{n_{\mathcal{X}_{im}} \times n_{\mathcal{X}_{i1}}} & \boldsymbol{A}_{i_m,i_1+1} & \cdots & \boldsymbol{A}_{i_m,i_m-1} & \boldsymbol{0}_{n_{\mathcal{X}_{im}} \times n_{\mathcal{X}_{im}}} & \boldsymbol{A}_{i_m,i_m+1} & \cdots & \boldsymbol{A}_{i_m,n} \end{bmatrix}}_{n \text{ columns}} \right\} \tag{14}$$

$$\tilde{\boldsymbol{A}}_i(2) = diag\left(\underbrace{\begin{bmatrix} \boldsymbol{0}_{n_{\mathcal{X}_i} \times n_{\mathcal{X}_1}} & \cdots & \boldsymbol{0}_{n_{\mathcal{X}_i} \times n_{\mathcal{X}_{i1}-1}} & \boldsymbol{A}_{i,i_1} & \boldsymbol{0}_{n_{\mathcal{X}_i} \times n_{\mathcal{X}_{i1}+1}} & \cdots & \boldsymbol{0}_{n_{\mathcal{X}_i} \times n_{\mathcal{X}_{im}-1}} & \boldsymbol{A}_{i,i_m} & \boldsymbol{0}_{n_{\mathcal{X}_i} \times n_{\mathcal{X}_{im}+1}} & \cdots & \boldsymbol{0}_{n_{\mathcal{X}_i} \times n_{\mathcal{X}_n}} \\ \boldsymbol{0}_{n_{\mathcal{X}_{i1}} \times n_{\mathcal{X}_1}} & \cdots & \boldsymbol{0}_{n_{\mathcal{X}_{i1}} \times n_{\mathcal{X}_{i1}-1}} & \boldsymbol{A}_{i_1,i_1} & \boldsymbol{0}_{n_{\mathcal{X}_{i1}} \times n_{\mathcal{X}_{i1}+1}} & \cdots & \boldsymbol{0}_{n_{\mathcal{X}_{i1}} \times n_{\mathcal{X}_{im}-1}} & \boldsymbol{A}_{i_1,i_m} & \boldsymbol{0}_{n_{\mathcal{X}_{i1}} \times n_{\mathcal{X}_{im}+1}} & \cdots & \boldsymbol{0}_{n_{\mathcal{X}_{i1}} \times n_{\mathcal{X}_n}} \\ \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ \boldsymbol{0}_{n_{\mathcal{X}_{im}} \times n_{\mathcal{X}_1}} & \cdots & \boldsymbol{0}_{n_{\mathcal{X}_{i1}} \times n_{\mathcal{X}_{i1}-1}} & \boldsymbol{A}_{i_m,i_1} & \boldsymbol{0}_{n_{\mathcal{X}_{i1}} \times n_{\mathcal{X}_{i1}+1}} & \cdots & \boldsymbol{0}_{n_{\mathcal{X}_{i1}} \times n_{\mathcal{X}_{im}-1}} & \boldsymbol{A}_{i_m,i_m} & \boldsymbol{0}_{n_{\mathcal{X}_{i1}} \times n_{\mathcal{X}_{im}+1}} & \cdots & \boldsymbol{0}_{n_{\mathcal{X}_i} \times n_{\mathcal{X}_n}} \end{bmatrix}}_{n \text{ columns}}, \; diag_{P-1}\left\{ \boldsymbol{0}_{\sum_{l \in \mathcal{N}_i^{\text{out}}} n_{\mathcal{X}_l} \times \sum_{l=1}^{n} n_{\mathcal{X}_l}} \right\} \right) \tag{15}$$

$$\tilde{\bar{\boldsymbol{B}}}_i = diag_P\left\{ \begin{bmatrix} \boldsymbol{B}_{i,1} & \cdots & \boldsymbol{B}_{i,i-1} & \boldsymbol{0}_{n_{\mathcal{X}_i} \times n_{u_i}} & \boldsymbol{B}_{i,i+1} & \cdots & \boldsymbol{B}_{i,n} \\ \boldsymbol{B}_{i_1,1} & \cdots & \boldsymbol{B}_{i_1,i-1} & \boldsymbol{0}_{n_{\mathcal{X}_{i1}} \times n_{u_i}} & \boldsymbol{B}_{i_1,i+1} & \cdots & \boldsymbol{B}_{i_1,n} \\ \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ \boldsymbol{B}_{i_m,1} & \cdots & \boldsymbol{B}_{i_m,i-1} & \boldsymbol{0}_{n_{\mathcal{X}_{im}} \times n_{u_i}} & \boldsymbol{B}_{i_m,i+1} & \cdots & \boldsymbol{B}_{i_m,n} \end{bmatrix} \right\} \tag{16}$$

$$\tilde{\boldsymbol{C}}_i = diag_P\left\{ \underbrace{\begin{bmatrix} \boldsymbol{C}_{i,1} & \cdots & \boldsymbol{C}_{i,i-1} & \boldsymbol{0}_{n_{\mathcal{Y}_i} \times n_{\mathcal{X}_i}} & \boldsymbol{C}_{i,i+1} & \cdots & \boldsymbol{C}_{i,i_1-1} & \boldsymbol{0}_{n_{\mathcal{Y}_i} \times n_{\mathcal{X}_{i1}}} & \boldsymbol{C}_{i,i_1+1} & \cdots & \boldsymbol{C}_{i,i_m-1} & \boldsymbol{0}_{n_{\mathcal{Y}_i} \times n_{\mathcal{X}_{im}}} & \boldsymbol{C}_{i,i_m+1} & \cdots & \boldsymbol{C}_{i,n} \\ \boldsymbol{C}_{i_1,1} & \cdots & \boldsymbol{C}_{i_1,i-1} & \boldsymbol{0}_{n_{\mathcal{Y}_{i1}} \times n_{\mathcal{X}_i}} & \boldsymbol{C}_{i_1,i+1} & \cdots & \boldsymbol{C}_{i_1,i_1-1} & \boldsymbol{0}_{n_{\mathcal{Y}_{i1}} \times n_{\mathcal{X}_{i1}}} & \boldsymbol{C}_{i_1,i_1+1} & \cdots & \boldsymbol{C}_{i_1,i_m-1} & \boldsymbol{0}_{n_{\mathcal{Y}_{i1}} \times n_{\mathcal{X}_{im}}} & \boldsymbol{C}_{i_1,i_m+1} & \cdots & \boldsymbol{C}_{i_1,n} \\ \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ \boldsymbol{C}_{i_m,1} & \cdots & \boldsymbol{C}_{i_m,i-1} & \boldsymbol{0}_{n_{\mathcal{Y}_{im}} \times n_{\mathcal{X}_i}} & \boldsymbol{C}_{i_m,i+1} & \cdots & \boldsymbol{C}_{i_m,i_1-1} & \boldsymbol{0}_{n_{\mathcal{Y}_{im}} \times n_{\mathcal{X}_{i1}}} & \boldsymbol{C}_{i_m,i_1+1} & \cdots & \boldsymbol{C}_{i_m,i_m-1} & \boldsymbol{0}_{n_{\mathcal{Y}_{im}} \times n_{\mathcal{X}_{im}}} & \boldsymbol{C}_{i_m,i_m+1} & \cdots & \boldsymbol{C}_{i_m,n} \end{bmatrix}}_{n \text{ columns}} \right\} \tag{17}$$

Author's personal copy

$$\boldsymbol{T}_i=\begin{bmatrix}\boldsymbol{0}_{(P-1)n_{\tilde{y}}\times n_{\tilde{y}}} & \boldsymbol{I}_{(P-1)n_{\tilde{y}}}\\ \boldsymbol{0}_{n_{\tilde{y}}\times(P-1)n_{\tilde{y}}} & \boldsymbol{I}_{n_{\tilde{y}}}\end{bmatrix},\quad n_x=\sum_{l=1}^{n}n_{x_l},\quad \overline{\boldsymbol{B}}_i=\begin{bmatrix} daig_M(\widehat{\boldsymbol{B}}_i)\\ \boldsymbol{0}_{n_{\tilde{x}_i}\times(M-1)n_{u_i}} \quad \widehat{\boldsymbol{B}}_i\\ \vdots \quad \vdots\\ \boldsymbol{0}_{n_{\tilde{x}_i}\times(M-1)n_{u_i}} \quad \widehat{\boldsymbol{B}}_i\end{bmatrix},$$

$$\overline{\boldsymbol{S}}_i=\begin{bmatrix}\widehat{\boldsymbol{A}}_i^0 & \cdots & \boldsymbol{0}\\ \vdots & \ddots & \vdots\\ \widehat{\boldsymbol{A}}_i^{P-1} & \cdots & \widehat{\boldsymbol{A}}_i^0\end{bmatrix},\quad \overline{\boldsymbol{C}}_i=diag_P\{\widehat{\boldsymbol{C}}_i\}.$$

The ND-MPC problem stated in Eq. (13) now can be formulated as a quadratic program by the introduction of the following matrices:

$$\boldsymbol{S}_i=\overline{\boldsymbol{C}}_i\overline{\boldsymbol{S}}_i,\quad \boldsymbol{N}_i=\boldsymbol{S}_i\overline{\boldsymbol{B}}_i\overline{\boldsymbol{\Gamma}}_i,$$

$$\underset{(M\text{ blocks})}{\boldsymbol{\Gamma}_i'}=\begin{bmatrix}\boldsymbol{I}_{n_{u_i}}\\ \vdots\\ \boldsymbol{I}_{n_{u_i}}\end{bmatrix},\quad \underset{(M\times M\text{ blocks})}{\overline{\boldsymbol{\Gamma}}_i}=\begin{bmatrix}\boldsymbol{I}_{n_{u_i}} & \cdots & 0\\ \vdots & \ddots & \vdots\\ \boldsymbol{I}_{n_{u_i}} & \cdots & \boldsymbol{I}_{n_{u_i}}\end{bmatrix},$$

$$\overline{\boldsymbol{Q}}_i=diag_P\{\widehat{\boldsymbol{Q}}_i\},$$

$$\overline{\boldsymbol{R}}_i=diag_P\{\boldsymbol{R}_i\}. \tag{21}$$

Lemma 3 (Quadratic program). *Under **Assumptions 1**, each controller $\mathcal{C}_i$, $i=1,\ldots,n$, has to solve at time k the following optimization problem*:

$$\min_{\Delta\boldsymbol{U}_i(k,M|k)}[\Delta\boldsymbol{U}_i^T(k,M|k)\boldsymbol{H}_i\Delta\boldsymbol{U}_i(k,M|k)-\boldsymbol{G}(k+1,P|k)\Delta\boldsymbol{U}_i(k,M|k)] \tag{22}$$

where the positive definite matrix $\boldsymbol{H}_i$ has the form

$$\boldsymbol{H}_i=\boldsymbol{N}_i^T\overline{\boldsymbol{Q}}_i\boldsymbol{N}_i+\overline{\boldsymbol{R}}_i \tag{23}$$

and

$$\boldsymbol{G}_i(k+1,P|k)=2\boldsymbol{N}_i^T\overline{\boldsymbol{Q}}_i[\boldsymbol{Y}_i^d(k+1,P|k)-\hat{\boldsymbol{Z}}_i(k+1,P|k)] \tag{24}$$

with

$$\begin{aligned}\hat{\boldsymbol{Z}}_i(k+1,P|k)=\boldsymbol{S}_i[&\overline{\boldsymbol{B}}_i\boldsymbol{\Gamma}_i'\boldsymbol{u}_i(k-1)+\overline{\boldsymbol{A}}_i^{(1)}\hat{x}(k|k)+\tilde{\boldsymbol{A}}_i\hat{\boldsymbol{X}}(k,P|k-1)\\ &+\tilde{\boldsymbol{B}}_i\boldsymbol{U}(k-1,M|k-1)]+\boldsymbol{T}_i\tilde{\boldsymbol{C}}_i\hat{\boldsymbol{X}}(k+1,P|k-1)\end{aligned} \tag{25}$$

Making use of these definitions

$$\boldsymbol{K}_i=\boldsymbol{\Gamma}_i\overline{\boldsymbol{K}}_i,\quad \boldsymbol{\Gamma}_i=\begin{bmatrix}\boldsymbol{I}_{n_{u_i}} & \boldsymbol{0}_{n_{u_i}\times Mn_{u_i}}\end{bmatrix},\quad \overline{\boldsymbol{K}}_i=\boldsymbol{H}_i^{-1}\boldsymbol{N}_i^T\overline{\boldsymbol{Q}}_i \tag{26}$$

The proof can be found in Appendix C. Based on Lemma 3, the following theorem can be deduced.

Theorem 1 (Closed-form solution). *Under **Assumptions 1**, for each controller $\mathcal{C}_i$, $i=1,\ldots,n$, the closed-form of the control law applied at time k at controller $\mathcal{C}_i$ to subsystem $\mathcal{S}_i$ is given by*

$$\boldsymbol{u}_i(k)=\boldsymbol{u}(k-1)+\boldsymbol{K}_i[\boldsymbol{Y}_i^d(k+1,P|k)-\hat{\boldsymbol{Z}}_i(k+1,P|k)] \tag{27}$$

The proof can be found in Appendix D.

Remark. The resulting complexity to obtain the closed-form solution for the local subsystem $\mathcal{S}_i$ is mainly given by the inversion of matrix $\boldsymbol{H}_i$. Considering that the size of matrix $\boldsymbol{H}_i$ equals $M\cdot n_{u_i}$, the complexity of the inversion algorithm is $\mathcal{O}(M^3,n_{u_i}^3)$ if using Gauss–Jordan algorithm. Therefore, the total computational complexity of the distributed solution is $\mathcal{O}(M^3,\sum_{i=1}^n n_{u_i}^3)$ while the computational complexity of the centralized MPC is $\mathcal{O}(M^3,(\sum_{i=1}^n n_{u_i})^3)$.

4. Stability analysis

On the basis of the closed-form solution stated by Theorem 1, the closed-loop dynamics can be specified and the stability condition can be verified by analyzing the closed-loop dynamic matrix. Thus, following theorem is obtained.

Theorem 2 (Neighbourhood-optimization distributed MPC stability). *The closed-loop system given by the feedback connection of plant $\mathcal{S}$ with the set of independent controller $\mathcal{C}_i$, $i=1,\ldots,n$, whose closed-form control laws are given by Eq. (28), is asymptotically stable if and only if*

$$|\lambda_j\{\boldsymbol{A}_N\}|<1,\forall j=1,\ldots,n_N \tag{28}$$

where $n_N=Pn_x+n_x+2Mn_u$ is the order of the global closed-loop system.

$$\boldsymbol{A}_N=\begin{bmatrix}\boldsymbol{A} & \boldsymbol{0} & \boldsymbol{B\Gamma} & \boldsymbol{0}\\ \boldsymbol{L\overline{S}A} & \boldsymbol{L\overline{S}\tilde{A}\Omega} & \boldsymbol{L\overline{S}\overline{B}} & \boldsymbol{L\overline{S}\tilde{B}\Pi}\\ \boldsymbol{\Theta A+\Phi L\overline{S}A} & \boldsymbol{\Phi L\overline{S}\tilde{A}\Omega} & \boldsymbol{\Theta B\Gamma+\Phi L\overline{S}\overline{B}+\Psi} & \boldsymbol{\Phi L\overline{S}\tilde{B}\Pi}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{Mn_u} & \boldsymbol{0}\end{bmatrix} \tag{29}$$

The proof can be found in Appendix E.

Remark. It should be noticed that the first two block rows of dynamic matrix $\boldsymbol{A}_N$ depend on element of matrix $\boldsymbol{A}$ (the first two block columns) and element of matrix $\boldsymbol{B}$ (in the last two block columns), while the third block row depends on process matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$, weight matrices $\boldsymbol{Q}_i$, $\boldsymbol{R}_i$ and horizons $\boldsymbol{P}$ and $\boldsymbol{M}$. This fact suggests a key for the design of ND-MPC. The degree of freedom available to the designer are on the choices of weight matrices $\boldsymbol{Q}_i$, $\boldsymbol{R}_i$ and horizons $\boldsymbol{P}$ and $\boldsymbol{M}$, which introduce significant modifications on the third block row of matrix $\boldsymbol{A}_N$.

5. Analysis of performance

To explain the essential differences between the optimization problem with neighbourhood optimization index and the optimization problem with local performance index, for each controller $\mathcal{C}_i$, $i=1,\ldots,n$, the optimization problem (13) of ND-MPC is rewritten into following form:

$$\begin{aligned}&\min_{\Delta\boldsymbol{U}_i(k,M|k)}\sum_{i=1}^{n}\left[\sum_{l=1}^{P}\|\hat{\boldsymbol{y}}_i(k+l|k)-\boldsymbol{y}_i^d(k+l|k)\|_{\boldsymbol{Q}_i}^2+\sum_{l=1}^{M}\|\Delta\boldsymbol{u}_i(k+l-1|k)\|_{\boldsymbol{R}_i}^2\right]\\ &\text{s.t.}\quad\begin{bmatrix}\hat{\boldsymbol{x}}_i(k+l+1|k)\\ \hat{\boldsymbol{x}}_{i_1}(k+l+1|k)\\ \vdots\\ \hat{\boldsymbol{x}}_{i_m}(k+l+1|k)\end{bmatrix}=\begin{bmatrix}\boldsymbol{A}_{ii} & \boldsymbol{A}_{ii_1} & \cdots & \boldsymbol{A}_{ii_m}\\ \boldsymbol{A}_{i_1i} & \boldsymbol{A}_{i_1i_1} & \cdots & \boldsymbol{A}_{i_1i_m}\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{A}_{i_mi} & \boldsymbol{A}_{i_mi_1} & \cdots & \boldsymbol{A}_{i_mi_m}\end{bmatrix}\begin{bmatrix}\hat{\boldsymbol{x}}_i(k+l|k)\\ \hat{\boldsymbol{x}}_{i_1}(k+l|k)\\ \vdots\\ \hat{\boldsymbol{x}}_{i_m}(k+l|k)\end{bmatrix}\\ &\qquad+\begin{bmatrix}\boldsymbol{B}_{ii}\\ \boldsymbol{B}_{i_1i}\\ \vdots\\ \boldsymbol{B}_{i_mi}\end{bmatrix}\boldsymbol{u}_i(k+l|k)+\hat{\hat{\boldsymbol{w}}}_i(k+l|k-1));\\ &\hat{\boldsymbol{x}}_j(k+l+1|k)=\hat{\boldsymbol{x}}_j(k+l+1\big|k-1),\quad (j\notin\mathcal{N}_i^{\text{out}});\\ &\hat{\boldsymbol{y}}_i(k+l|k)=\boldsymbol{C}_i\hat{\boldsymbol{x}}_i(k+l|k)+\hat{\boldsymbol{v}}_i(k+l|k-1),\quad (i=1,\ldots,n);\\ &\Delta\boldsymbol{u}_j(k+l-1|k)=\Delta\boldsymbol{u}_j(k+l-1|k-1),\quad (j\neq i).\end{aligned} \tag{30}$$

If using the local performance index (4), as in Vaccarini, Longhi, and Katebi (2009), for each controller $\mathcal{C}_i$, $i=1,\ldots,n$, the optimization problem of distributed MPC can be written into following form:

$$\min_{\Delta\boldsymbol{U}_i(k,M|k)}\sum_{j=1}^{n}\left[\sum_{l=1}^{P}\|\hat{\boldsymbol{y}}_j(k+l|k)-\boldsymbol{y}_j^d(k+l|k)\|_{\boldsymbol{Q}_j}^2+\sum_{l=1}^{M}\|\Delta\boldsymbol{u}_j(k+l-1|k)\|_{\boldsymbol{R}_j}^2\right]$$

Author's personal copy

$$\text{s.t.}\quad \hat{x}_i(k+l+1|k)=\boldsymbol{A}_{ii}x_i(k+l|k)+\boldsymbol{B}_{ii}u_i(k+l|k)+\hat{\boldsymbol{w}}_i(k+l|k-1));$$
$$\hat{\boldsymbol{y}}_i(k+l|k)=\widehat{\boldsymbol{C}}_i\hat{x}_i(k+l|k)+\hat{\widehat{\boldsymbol{v}}}_i(k+l|k-1);$$
$$\hat{\boldsymbol{y}}_j(k+l|k)=\hat{\boldsymbol{y}}_j(k+l|k-1)\quad (j\neq i);$$
$$\Delta\boldsymbol{u}_j(k+l-1|k)=\Delta\boldsymbol{u}_j(k+l-1|k-1)\quad (j\neq i). \tag{31}$$

It can be seen that the performance indices of problem (30) and (31) are same to each other. In neighbourhood optimization, the state evolutions of subsystem $\mathcal{S}_i$ and its output-neighbours are solved together, the impact of control decision $\Delta\boldsymbol{U}_i(k,M|k)$ on the states of $\mathcal{S}_i$ and its output-neighbours is fully considered. However in problem (31), only the state evolution of subsystem $\mathcal{S}_i$ is determined by $\Delta\boldsymbol{U}_i(k,M|k)$ and the states of other subsystems are substituted by the estimations at time $k-1$. It is clearly that the predictive model in problem (30) is more close to system model (2). Thus, it is more reasonable to adopt neighbourhood performance index rather than to use local performance index.

In fact, after several control periods, the control decision $\Delta\boldsymbol{U}_i(k,M|k)$ affects not only the output-neighbours of $\mathcal{S}_i$ but also other subsystems (e.g. the output-neighbours of the output-neighbours of $\mathcal{S}_i$). Here, the interactions with other subsystems except output-neighbours are neglected. If there is enough network band-width for employing iterative algorithm, these interactions can also be taken into account.

It should be noticed that each controller only communicates with its neighbours and its neighbours' neighbours in ND-MPC. Moreover, if each controller communicates with its neighbours twice within a sampling time interval, the information of its neighbours' neighbours can be obtained from its neighbours. That means only the information exchanging among neighbourhood is required using this method. Thus, if one subsystem fails, the other subsystem unrelated to $\mathcal{S}_i$ can be run normally. The communication loads related to $\mathcal{S}_i$ are that $\mathcal{S}_i$ get its future states to its neighbours and sent its neighbour's states and inputs to its neighbours. That means the maximum communication loads is

$$\sum_{i=1}^{n} m_i\left(Pn_{xi}+\sum_{j\in\mathcal{N}_i}Pn_{xj}\right) \tag{32}$$

In Eq. (32) some information are calculated repeatedly, thus (32) is the maximum communication burden using ND-MPC. The required physical connections equals to the number of none zero elements in $\boldsymbol{A}$ subtract the rank of $\boldsymbol{A}$.

Since the computational burden mainly comes from the complexity of the inversion algorithm, see remark in Section 3, the computational burden is similar to (a little more than) that of the method proposed in Vaccarini, Longhi, and Katebi (2009). The memory required is a little larger than that in Vaccarini, Longhi, and Katebi (2009) since the system matrices dimension of each subsystem's states evolution equation is larger than that in Vaccarini, Longhi, and Katebi (2009). However the memory is not a problem with the modern computer technology. The efficiency of this method will be validated in next section.

6. Experiment validation

To illustrate the performance of proposed method, application of this method to accelerated cooling process (ACC) test rig is performed in one steel company in Shanghai, China.

6.1. Process description

ACC process, simplified in Fig. 2, is used to cool a metal plate from initial temperature around 750–800 °C down to final temperature in the range of 450–560 °C. A constant cooling curve of plate is required in ACC, which helps a lot to strongly improve the mechanical characteristics of the corresponding products. The cooling area is partitioned into three sections: air cooling section, water cooling section and re-reddening section, labelled A, B and C, respectively. Fifteen cooling header units are uniformly spaced along section B. The number of cooling header units in operation (N), the water flux of each cooling unit (F) can be adjusted separately. The temperature drop is caused by the heat radiation in sections A and C, and caused by both radiation and water cooling in section B (Guan, Wang, & Chai, 1998; Mukhopadhyay & Sikdar, 2005). Four pyrometers $T_{P1}\sim T_{P4}$ are located in the positions of 13.5, 58.6, 89m and 109.5 m, respectively. The temperatures of plate inside cooling section are measured by soft-sensors.

The control objective is to control the location-dependent temperatures at location $l_1,l_2,\ldots,l_n$ to be consistent with the reference temperature denoted with $\boldsymbol{y}^d=\begin{bmatrix}y_1^d & y_2^d & \cdots & y_n^d\end{bmatrix}^T$ through adjusting the flux of each water cooling header unit and plate velocity. (l_0 is the location of T_{P1}; l_i, $i=1,2,\ldots,15$, is the entry of ith cooling header unit; l_{n-1} is the exit of section B; and l_n is the location of T_{P3}.)

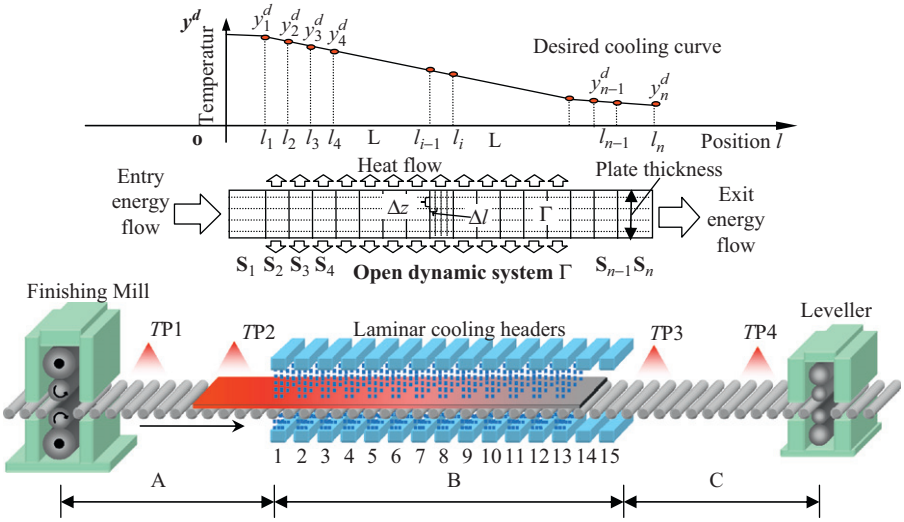


Fig. 2. ACC process for middle and heavy plate.

Author's personal copy

6.2. Test rig and control system

The test rig and the automation system structure of the test rig is show in Fig. 3. The automation system consists of six industrial personal computers (IPC), one PLC, and many I/O modules. The software of WinCC and an OPC sever running on IPC 1 are used to monitor the cooling process, and the PI controllers running on PLC are employed to control the water flux. The advanced control algorithm is running on IPC2–IPC6, which communicate with PLC through the OPC server running on IPC 1.

6.3. System state space model

Consider the overall system of the cooling area from the point of view of geometrically distributed setting system, the geometrical locations of temperature sensor T_{P2} and T_{P3} as well as the plate top and bottom sides represent an open thermodynamic system $\boldsymbol{\Gamma}$. The overall system, therefore, can be dismembered into n subsystems in accordance with the location coordinate l. The sth subsystem ranges from l_{s-1} to l_s $(s=1,2,\ldots,n)$ as shown in Fig. 2. The input of sth subsystem is the water flux of corresponding header and the output is the plate temperature at location l_s.

For the numeric treatment of heat conductivity through plate thickness, the system $\boldsymbol{\Gamma}$ is broken into m layers, and each subsystem is divided into n_s volumes in l-direction. Denote the temperature of ith in z-direction and jth in l-direction volume with $x_s^{(i,j)}$ and set the sampling time be Δt. Then following linear state space representation of subsystem $\mathcal{S}_s$ can be deduced (Zheng, Li, & Wang, 2009)

$$\begin{cases} \boldsymbol{x}_s(k+1)=\boldsymbol{A}_{ss}\cdot\boldsymbol{x}_s(k)+\boldsymbol{B}_{ss}\cdot u_s(k)+\boldsymbol{D}_{s,s-1}\cdot\boldsymbol{x}_{s-1}(k) \\ y_s(k)=\boldsymbol{C}_{ss}\cdot\boldsymbol{x}_s(k) \end{cases} \quad s=1,2,\ldots,N \tag{33}$$

where $\boldsymbol{x}_s=[(\boldsymbol{x}_{s,1})^T \quad (\boldsymbol{x}_{s,2})^T \quad \cdots \quad (\boldsymbol{x}_{s,n_s})^T]^T$, $\boldsymbol{x}_{s,j}=[x_s^{(1,j)}\ x_s^{(2,j)}\cdots x_s^{(m,j)}]^T$, $j=1,2,\ldots,n_s$ is the state vector of subsystem $\mathcal{S}_s$, y_s is average temperature of the last column volumes of subsystem $\mathcal{S}_s$, u_s is the input of subsystem $\mathcal{S}_s$ and there is a fixed relationship between u_s and the water flux in subsystem $\mathcal{S}_s$. $\boldsymbol{A}_{ss}$, $\boldsymbol{B}_{ss}$, $\boldsymbol{D}_{s,s-1}$ and $\boldsymbol{C}_{ss}$ are coefficient matrices of subsystem $\mathcal{S}_s$ with

$$\boldsymbol{A}_{ss}=\begin{bmatrix} \boldsymbol{\Phi}_s^{(1)}\cdot\boldsymbol{\Lambda} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Phi}_s^{(2)}\cdot\boldsymbol{\Lambda} & & \vdots \\ \vdots & & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{\Phi}_s^{(n_s)}\cdot\boldsymbol{\Lambda} \end{bmatrix} + \begin{bmatrix} (1-\gamma)\boldsymbol{I}_m & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \gamma\boldsymbol{I}_m & (1-\gamma)\boldsymbol{I}_m & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \gamma\boldsymbol{I}_m & (1-\gamma)\boldsymbol{I}_m \end{bmatrix}$$

$$\boldsymbol{B}_{ss}=\begin{bmatrix} \boldsymbol{\psi}_s^{(1)} \\ \vdots \\ \boldsymbol{\psi}_s^{(n_s)} \end{bmatrix}, \quad \boldsymbol{C}_{ss}=m^{-1}\cdot\begin{bmatrix} \boldsymbol{0}^{1\times m(n_s-1)} & \boldsymbol{1}^{1\times m} \end{bmatrix},$$

$$\boldsymbol{D}_{s,s-1}=\begin{bmatrix} \boldsymbol{0}^{m\times m(n_s-1)} & \gamma\boldsymbol{I}_m \\ \boldsymbol{0}^{m(n_s-1)\times m(n_s-1)} & \boldsymbol{0}^{m(n_s-1)\times m} \end{bmatrix}$$

and

$$\boldsymbol{\Phi}_s^{(j)}=\begin{bmatrix} a(\check{x}_s^{(1,j)}) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & a(\check{x}_s^{(m,j)}) \end{bmatrix};$$

$$\boldsymbol{\psi}_s^{(j)}(\boldsymbol{x}_s)=\begin{bmatrix} \theta_s^{(1,j)}(\check{x}_s^{(1,j)}-x_\infty)\beta(\check{x}_s^{(1,j)}) \\ \boldsymbol{0}^{(m-2)\times 1} \\ \theta_s^{(m,j)}(\check{x}_s^{(m,j)}-x_\infty)\beta(\check{x}_s^{(m,j)}) \end{bmatrix};$$

$$\boldsymbol{\Lambda}=\begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & -2 & 1 \\ 0 & \cdots & 0 & 1 & -1 \end{bmatrix};$$

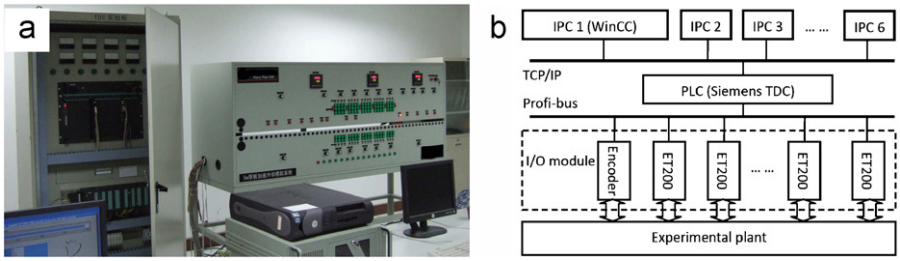


Fig. 3. The test rig: (a) ACC pilot apparatus and (b) the automation system.

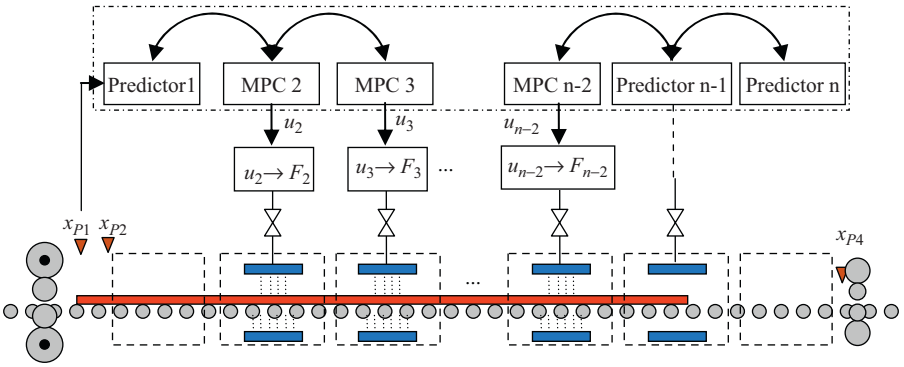


Fig. 4. Control strategy of ACC.

Author's personal copy

$$\boldsymbol{I}_m \in \mathbb{R}^{m\times m};\quad \begin{cases}\theta_s^{(i,j)} = (\breve{x}_s^{(i,j)}/x)^a, & s\in\mathcal{C}_{\mathcal{W}}\\ \theta_s^{(i,j)} = h_{air}(\breve{x}_s^{(i,j)}), & s\in C_{\mathcal{A}}\end{cases}$$

$$\begin{cases}u_s = 2186.7\times10^{-6}\times a(v/v_0)^b\times(F_s/F_0)^c, & s\in\mathcal{C}_{\mathcal{W}}\\ u_s = 1, \quad s\in\mathcal{C}_{\mathcal{A}}\end{cases}$$

$$a(x_s^{(i,j)}) = -\Delta t\lambda(x_s^{(i,j)})/(\Delta z^2\rho(x_s^{(i,j)})c_p(x_s^{(i,j)})),$$
$$\beta(x_s^{(i,j)}) = \Delta t a(x_s^{(i,j)})/\lambda(x_s^{(i,j)})$$
$$\gamma = \Delta t v/\Delta l,\quad i=1,2,\ldots,m,\quad j=1,2,\ldots,n_s$$

where Δl and Δz are the longitude and thickness of each volume, ρ is the plate density, c_p is the specific heat capacity, λ is the heat conductivity, v is the plate velocity and $\breve{x}_s^{(i,j)}$ is the equilibrium state of $\mathcal{S}_s$. $\mathcal{C}_{\mathcal{W}}$ is the set of subsystems, in which plate is cooled by water. $\mathcal{C}_{\mathcal{A}}$ is the set of subsystems in which plate is cooled major through radiation, F_s is the water flux of the header unit in $\mathcal{S}_s$, F_0, v_0, a, b and c are constants, and their detailed definitions are available in Zheng, Li, and Wang (2009, 2010). The different value of ρ, c_p and λ with different steel temperature can be also found in Zheng, Li, and Wang (2010).

6.4. Control strategy

With the development of computer technologies and control theory, many advanced control methodologies have been successfully applied to complex metallurgical processes (Jin, Zhou, & Chang, 2008; Tang, Wang, & Liu, 2008; Wang, Wu, & Chai, 2004; Zhou, Chai, & Wang, 2009). As for ACC process, the proposed ND-MPC is adopted in this work. As shown in Fig. 4, each subsystem is controlled by a local MPC. As for the subsystems in which the corresponding cooling water header unit is closed, the local MPC is substituted with a predictor. The predictor estimates the future states of corresponding subsystem and broadcasts the estimations to its neighbours.

Table 2
The plate parameters and the operating points.

Item	Value
Thickness of plate	19.28 mm
Length of plate	25 m
Environment temperature	25 °C
Starting temperature	750–800 °C
Desired final temperature	510–550 °C
Average velocity of plate	1.6 m/s
Number of header opened	12
Sampling period	0.37 s

6.5. Performance of system

One X70 pipe steel plate is taken as an example. The parameters of this steel plate are listed in Table 2. And the equilibriums temperature of this plate is shown in Fig. 5.

Set both the prediction horizon (P) and control horizon (M) of each local MPC equal to 10. And set the weights of outputs and inputs equal to 1 in the optimization index. Set the starting cooling temperature (T_{P2}) in whole process be 780 °C. The resulting performance of closed-loop system using centralized MPC, ND-MPC and networked decentralized MPC described in Vaccarini, Longhi, and Katebi (2009) are presented in Fig. 6, and the corresponding manipulated variables (unit: 1 m^{-2} min^{-1}) are shown in Fig. 3.

It can is observed from Figs. 6 and 7 that the performance of closed-loop system using ND-MPC is improved significantly comparing with that using the networked decentralized MPC proposed in Vaccarini, Longhi, and Katebi (2009). Both the control decision and performance of closed-loop system using ND-MPC are very close to those using centralized MPC. Furthermore, there is less computation demand using the ND-MPC than using centralized MPC. Thus, the ND-MPC is an effective method which could guarantee global performance improvement with fast computational speed and less communication burden.

7. Conclusions

In this paper, the control for a class of large scale system which is naturally divided into many small scale interacting subsystems is discussed. A novel distributed MPC framework based on

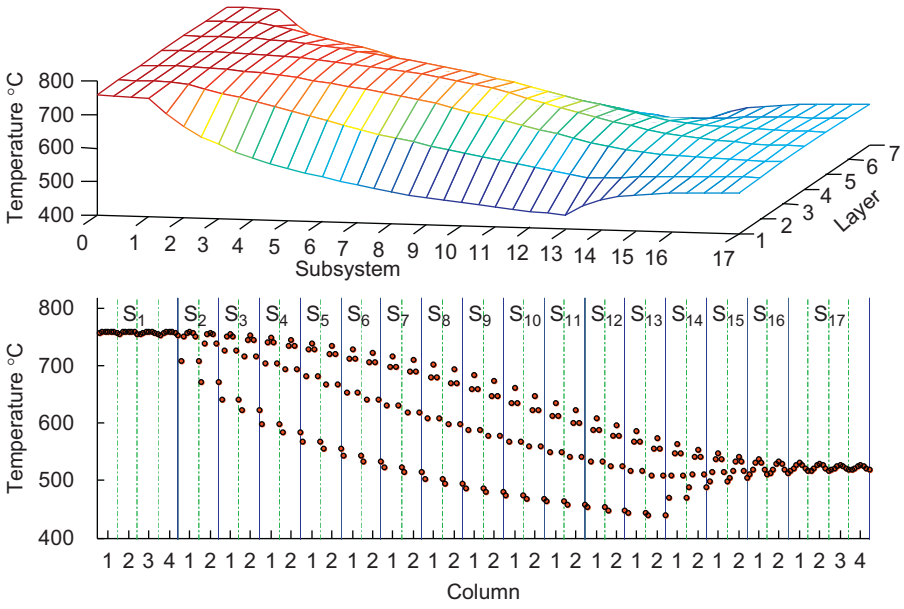


Fig. 5. Equilibriums of states of entire system.

Author's personal copy

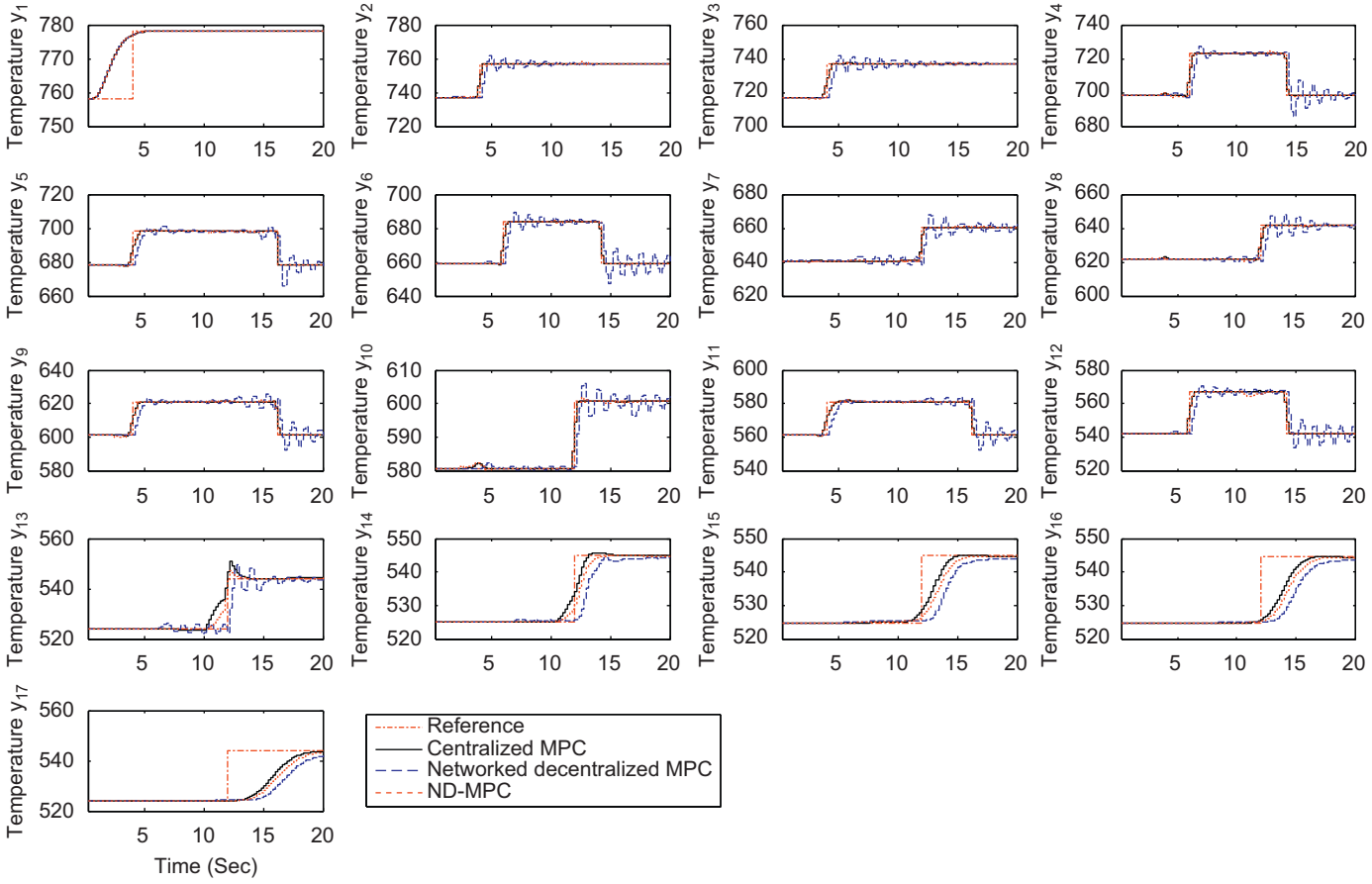


Fig. 6. Performance of close-loop subsystems using centralized MPC, ND-MPC and the Networked decentralized MPC.

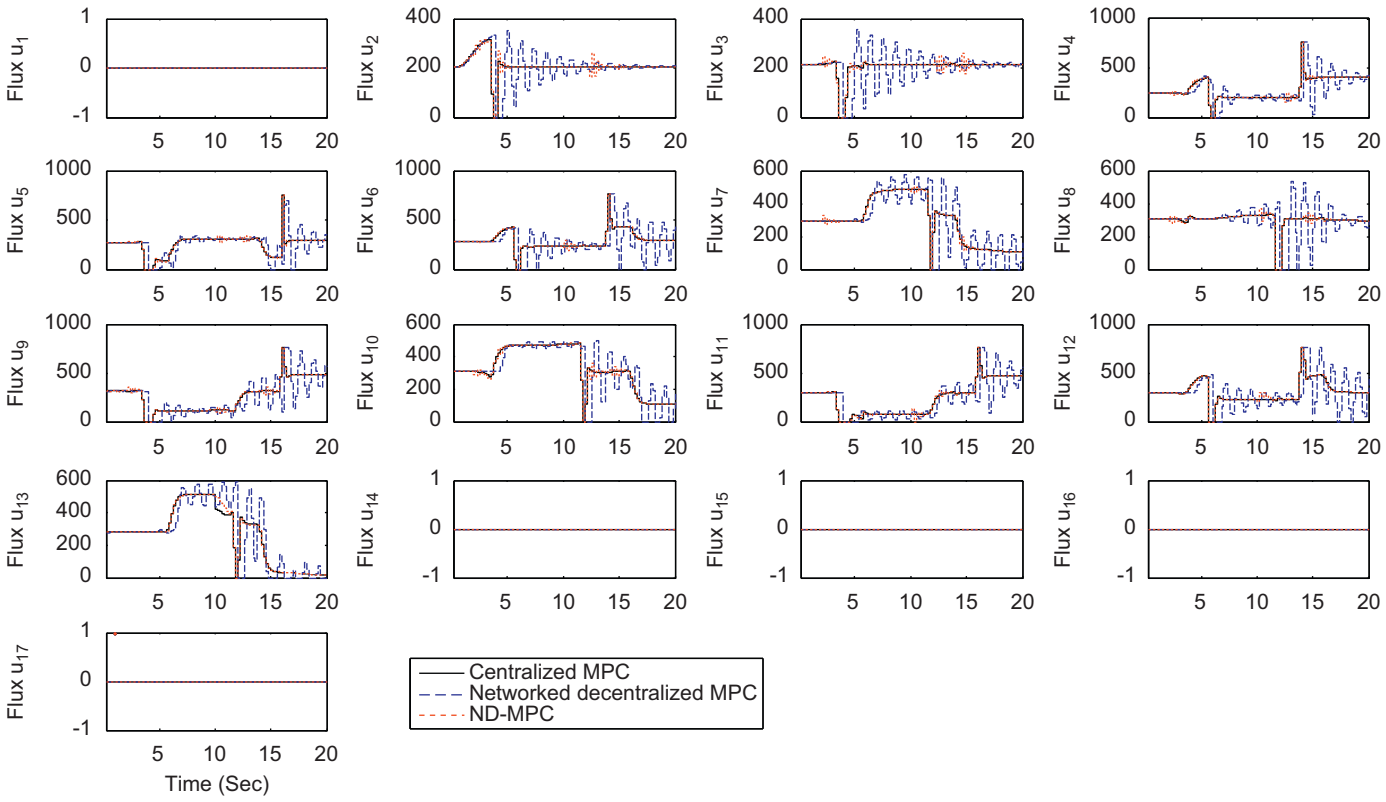


Fig. 7. Flux of each header unit using Centralized MPC, ND-MPC and Networked decentralized MPC.

Author's personal copy

neighbourhood optimization is present and the condition of closed-loop stability is given for local MPCs tuning. In the procedure of resolving optimal solution, each subsystem only communicates with its neighbours, which is rather easy to fulfil the network requirements. Moreover, the discussion of the performance of proposed methodology and the application of ND-MPC to ACC test rig prove that the proposed method guarantees an improving performance of entire system with relative relaxed communication requirements. Further investigation will focus on designing stable distributed MPC with constraints and global performance improvement for this class of large scale systems.

Acknowledgement

This work was supported by the National Nature Science Foundation of China (Grants: 60825302, 60934007, 61074061), the High Technology Research and Development Program of China (Grant: 2009AA04Z162), and partly sponsored by the Program of Shanghai Subject Chief Scientist, and "Shu Guang" project supported by Shanghai Municipal Education Commission and Shanghai Education Development Foundation, and the Key Project of Shanghai Science and Technology Commission (10JC1403400). The authors would like to think the anonymous referees and the Associate Editor for their constructive and valuable comments that have improved the manuscript.

Appendix A. Proof of Lemma 1

The proof is stated by writing, for $i=1,\ldots,n$, the h-ahead predictions at time k based on the information computed at time $k-1$ of the interaction vectors (9) and (10) and by representing them in a stacked form for $h=1,\ldots,P$. The last $P-M-1$ samples of the stacked control action predictions $\boldsymbol{U}_j(k,P|k-1)(j=1,2,\ldots,n)$, that are not contained in $\boldsymbol{U}_j(k-1,M|k-1)$, are assumed equal to the last element of $\boldsymbol{U}_j(k-1,M|k-1)$. By definitions (14)–(18) and Table 1, this implies that relations (19) hold.

Appendix B. Proof of Lemma 2

By Eqs. (11) and (12), and imposing that $\boldsymbol{u}_i(k+P-1|k)=\boldsymbol{u}_i(k+P-2|k)=\cdots=\boldsymbol{u}_i(k+M|k)=\boldsymbol{u}_i(k+M-1|k)$ and $\hat{\boldsymbol{v}}_i(k+P|k-1)=\hat{\boldsymbol{v}}_i(k+P-1|k-1)$, also substituting $\hat{\boldsymbol{W}}_i(k,p|k-1)$ and $\hat{\boldsymbol{V}}_i(k,p|k-1)$ with their explicit expressions (19), it results the following stacked state prediction for controller $\mathcal{C}_i$:

$$\begin{aligned}\hat{\boldsymbol{X}}_i(k+1,P|k)=\overline{\boldsymbol{S}}_i[\overline{\boldsymbol{A}}_i\hat{\boldsymbol{x}}_i(k|k)+\overline{\boldsymbol{B}}_i\boldsymbol{U}_i(k,M|k)+\tilde{\boldsymbol{A}}_{i1}\hat{\boldsymbol{X}}(k,p|k)\\+\tilde{\boldsymbol{B}}_i\boldsymbol{U}(k-1,M|k-1)].\end{aligned}$$

Let $\hat{\boldsymbol{x}}_i'(k|k-1)=\left[\hat{\boldsymbol{x}}_{i_1}^T(k|k-1)\quad\cdots\quad\hat{\boldsymbol{x}}_{i_{mi}}^T(k|k-1)\right]^T$, and by definitions (8), (14) and (15), the above equation becomes

$$\begin{aligned}\hat{\boldsymbol{X}}_i(k+1,P|k)&=\overline{\boldsymbol{S}}_i[\overline{\boldsymbol{A}}_i^{(1)}\hat{\boldsymbol{x}}(k|k)+\overline{\boldsymbol{A}}_i^{(2)}\hat{\boldsymbol{x}}_i'(k|k-1)+\overline{\boldsymbol{B}}_i\boldsymbol{U}_i(k,M|k)\\&\quad+\tilde{\boldsymbol{A}}_i^{(1)}\hat{\boldsymbol{X}}(k,P|k-1)+\tilde{\boldsymbol{B}}_i\boldsymbol{U}(k-1,M|k-1)]\\&=\overline{\boldsymbol{S}}_i[\overline{\boldsymbol{A}}_i^{(1)}\hat{\boldsymbol{x}}(k|k)+\overline{\boldsymbol{B}}_i\boldsymbol{U}_i(k,M|k)+(\tilde{\boldsymbol{A}}_i^{(}1)+\tilde{\boldsymbol{A}}_i^{(2)})\hat{\boldsymbol{X}}(k,P|k-1)\\&\quad+\tilde{\boldsymbol{B}}_i\boldsymbol{U}(k-1,M|k-1)]\\&=\overline{\boldsymbol{S}}_i[\overline{\boldsymbol{A}}_i^{(1)}\hat{\boldsymbol{x}}(k|k)+\overline{\boldsymbol{B}}_i\boldsymbol{U}_i(k,M|k)+\tilde{\boldsymbol{A}}_i\hat{\boldsymbol{X}}(k,P|k-1)\\&\quad+\tilde{\boldsymbol{B}}_i\boldsymbol{U}(k-1,M|k-1)]\end{aligned}$$

By model (7) and definitions of the coefficients in (20), the stacked output prediction for controller $\mathcal{C}_i$ can be expressed as

$$\hat{\boldsymbol{Y}}_i(k+1,P|k)=\overline{\boldsymbol{C}}_i\hat{\boldsymbol{X}}_i(k+1,P|k)+\boldsymbol{T}_i\tilde{\boldsymbol{C}}_i\hat{\boldsymbol{X}}(k+1,P|k-1).$$

This proves the Lemma 2.

Appendix C. Proof of Lemma 3

Making use of stacked vectors and definitions (21), the cost function (6) to be minimized by controller $\mathcal{C}_i$ can be expressed in the equivalent form

$$\bar{J}_i=\|\hat{\boldsymbol{Y}}_i(k+1,P|k)-\hat{\boldsymbol{Y}}_i^d(k+1,P|k)\|_{\overline{\boldsymbol{Q}}_i}^2+\|\Delta\boldsymbol{U}_i(k,M|k)\|_{\overline{\boldsymbol{R}}_i}^2$$

The stacked local output prediction $\hat{\boldsymbol{Y}}_i(k+1,P|k)$ is a function of the control action, therefore, in order to express $\bar{J}_i$ as a function of the control sequence $\Delta\boldsymbol{U}_i(k,M|k)$, an explicit expression for such a prediction is needed. Considering that $\boldsymbol{u}_i(k+h|k)=\boldsymbol{u}_i(k-1)+\sum_{r=0}^{h}\Delta\boldsymbol{u}_i(k+r|k)$, $h=1,2,\ldots,M$, the local stacked control sequence $\boldsymbol{U}_i(k,M|k)$ is used together with (20) and (21) to obtain the output prediction in the form $\hat{\boldsymbol{Y}}_i(k+1,P|k)=\boldsymbol{N}_i\Delta\boldsymbol{U}_i(k,M|k)+\hat{\boldsymbol{Z}}(k+1,P|k)$. By substituting this expression, the local cost function $\bar{J}_i$ takes the form (22). The positive definiteness of matrices $\overline{\boldsymbol{Q}}_i$ and $\overline{\boldsymbol{R}}_i$ implies the same property for matrix $\boldsymbol{H}_i$.

In this way the ND-MPC problem has been transformed into an equivalent unconstrained QP problem which has to be locally solved online at each sampling instant.

Appendix D. Proof of Theorm 1

States that a solution to ND-MPC problem minimizes cost function (22) with respect to the control sequence $\Delta\boldsymbol{U}_i(k,M|k)$. This solution has the form $\Delta\boldsymbol{U}_i(k,M|k)=((1/2)\boldsymbol{H}_i^{-1}\boldsymbol{G}_i(k+1,P|k))$. Following the receding horizon strategy, only the first element of the optimal sequence is actually applied to the process and the control action is expressed as $\boldsymbol{u}_i(k)=\boldsymbol{u}_i(k-1)+\boldsymbol{\Gamma}_i\Delta\boldsymbol{U}_i(k,M|k)$ which gives the final closed-form (27).

Appendix E. Proof of Theorm 2

To simplify the process of stability proof, define that

$$\begin{aligned}&\boldsymbol{\Omega}=\left[\boldsymbol{\Omega}_1^T\quad\cdots\quad\boldsymbol{\Omega}_P^T\right]^T,\quad\boldsymbol{\Omega}_j=diag\{\boldsymbol{\Omega}_{1j},\ldots,\boldsymbol{\Omega}_{nj}\},\\&\boldsymbol{\Omega}_{ij}=\left[\boldsymbol{0}_{n_{x_i}\times(j-1)n_{x_i}}\quad\boldsymbol{I}_{n_{x_i}}\quad\boldsymbol{0}_{n_{x_i}\times(P-j)n_{x_i}}\right],\quad(i=1,\ldots,n,\quad j=1,\ldots,P);\end{aligned}\tag{D.1}$$

$$\begin{aligned}&\boldsymbol{\Pi}=\left[\boldsymbol{\Pi}_1^T\quad\cdots\quad\boldsymbol{\Pi}_M^T\right]^T,\quad\boldsymbol{\Pi}_j=diag\{\boldsymbol{\Pi}_{1j},\ldots,\boldsymbol{\Pi}_{nj}\},\\&\boldsymbol{\Pi}_{ij}=\left[\boldsymbol{0}_{n_{u_i}\times(j-1)n_{u_i}}\quad\boldsymbol{I}_{n_{u_i}}\quad\boldsymbol{0}_{n_{u_i}\times(M-j)n_{u_i}}\right],\quad(i=1,\ldots,n,\quad j=1,\ldots,M).\end{aligned}\tag{D.2}$$

The following equations are achieved:

$$\hat{\boldsymbol{X}}(k,P|k-1)=\boldsymbol{\Omega}\hat{\mathbb{X}}(k,P|k-1)\tag{D.3}$$

$$\boldsymbol{U}(k,M|k-1)=\boldsymbol{\Pi}\boldsymbol{U}(k,M|k-1)\tag{D.4}$$

Define

$$\overline{\boldsymbol{A}}=diag\{\overline{\boldsymbol{A}}_{11},\ldots,\overline{\boldsymbol{A}}_{n1}\};\quad\tilde{\boldsymbol{A}}=\left[\tilde{\boldsymbol{A}}_1^T\cdots\quad\tilde{\boldsymbol{A}}_n^T\right]^T;\ \overline{\boldsymbol{B}}=diag\{\overline{\boldsymbol{B}}_1,\ldots,\overline{\boldsymbol{B}}_n\};$$

$$\tilde{\boldsymbol{B}}=\left[\tilde{\boldsymbol{B}}_1^T\cdots\quad\tilde{\boldsymbol{B}}_n^T\right]^T;\quad\boldsymbol{L}=diag\{\boldsymbol{L}_1,\ldots,\boldsymbol{L}_n\};$$

Author's personal copy

$$\boldsymbol{L}_i = diag_P\left\{\left[\begin{array}{cc}\boldsymbol{I}_{n_{x_i}} & \boldsymbol{0}_{n_{x_i}\times(n_{\tilde{x}_i}-n_{x_i})}\end{array}\right]\right\}; \qquad \overline{\boldsymbol{S}} = diag\{\overline{\boldsymbol{S}}_1,\ldots,\overline{\boldsymbol{S}}_n\}; \tag{D.5}$$

Then, for each controller $\mathcal{C}_i$, $i=1,\ldots,n$, by Lemma 2 and definitions (D.5), the stacked distributed state prediction at time k are expressed by

$$\begin{aligned}\hat{\boldsymbol{X}}_i(k+1,P|k) &= \boldsymbol{L}_i\hat{\overline{\boldsymbol{X}}}_i(k+1,P|k) = \boldsymbol{L}_i\overline{\boldsymbol{S}}_i[\overline{\boldsymbol{A}}_{i1}\hat{x}(k|k)+\overline{\boldsymbol{B}}_i\boldsymbol{U}_i(k,M|k)\\ &\quad+\tilde{\boldsymbol{A}}_i\hat{\boldsymbol{X}}(k,P|k-1)+\tilde{\boldsymbol{B}}_i\boldsymbol{U}(k-1,M|k-1)]\end{aligned} \tag{D.6}$$

By definitions (D.5), the completed stacked distributed prediction can be expressed as

$$\begin{aligned}\hat{\mathbb{X}}(k+1,P|k) &= \boldsymbol{L}\overline{\boldsymbol{S}}[\overline{\boldsymbol{A}}\hat{x}(k|k)+\overline{\boldsymbol{B}}\mathbb{U}(k,M|k)+\tilde{\boldsymbol{A}}\hat{\boldsymbol{X}}(k,P|k-1)\\ &\quad+\tilde{\boldsymbol{B}}\boldsymbol{U}(k-1,M|k-1)]\end{aligned} \tag{D.7}$$

Substituting (D.3) and (D.2) into (D.7), the following complete version of the stacked distributed prediction can be deduced:

$$\begin{aligned}\hat{\mathbb{X}}(k+1,P|k) &= \boldsymbol{L}\overline{\boldsymbol{S}}[\overline{\boldsymbol{A}}\hat{x}(k|k)+\overline{\boldsymbol{B}}\mathbb{U}(k,M|k)+\tilde{\boldsymbol{A}}\boldsymbol{\Omega}\hat{\mathbb{X}}(k,P|k-1)\\ &\quad+\tilde{\boldsymbol{B}}\boldsymbol{\Pi}\hat{\mathbb{U}}(k-1,M|k-1)]\end{aligned} \tag{D.8}$$

Considering that the local control action applied at time $k-1$ is given by $\boldsymbol{u}_i(k-1)=\boldsymbol{\Gamma}_i\boldsymbol{U}_i(k-1,m|k-1)$, the open-loop optimal sequence $\boldsymbol{U}_i(k,M|k)$ of controller $\mathcal{C}_i$ at time k can be expressed as $\boldsymbol{U}_i(k,M|k)=\boldsymbol{\Gamma}_i'\boldsymbol{\Gamma}_i\boldsymbol{U}_i(k-1,M|k-1)+\overline{\boldsymbol{\Gamma}}_i\Delta\boldsymbol{U}_i(k,M|k)$. Then by Eqs. (26) and (28), the stacked open-loop optimal control sequence at time k can be directly expressed as

$$\begin{aligned}\boldsymbol{U}_i(k,M|k-1) &= \boldsymbol{\Gamma}_i'\boldsymbol{u}_i(k-1)+\overline{\boldsymbol{\Gamma}}_i\overline{\boldsymbol{K}}_i[\boldsymbol{Y}_i^d(k+1,P|k)-\hat{\boldsymbol{Z}}_i(k+1,P|k)]\\ &= \boldsymbol{\Gamma}_i'\boldsymbol{u}_i(k-1)+\overline{\boldsymbol{\Gamma}}_i\overline{\boldsymbol{K}}_i\{\boldsymbol{Y}_i^d(k+1,P|k)-\boldsymbol{S}_i[\overline{\boldsymbol{B}}_i\boldsymbol{\Gamma}_i'\boldsymbol{u}_i(k-1)\\ &\quad+\overline{\boldsymbol{A}}_i^{(1)}\hat{x}(k|k)+\tilde{\boldsymbol{A}}_i\hat{\boldsymbol{X}}(k,P|k-1)+\tilde{\boldsymbol{B}}_i\boldsymbol{U}(k-1,M|k-1)]\\ &\quad-\boldsymbol{T}_i\tilde{\boldsymbol{C}}_i\hat{\boldsymbol{X}}(k,P|k-1)\}\end{aligned} \tag{D.9}$$

Define that

$$\begin{aligned}&\boldsymbol{\Gamma}' = diag\{\boldsymbol{\Gamma}_1',\ldots,\boldsymbol{\Gamma}_n'\}, && \boldsymbol{\Gamma} = diag\{\boldsymbol{\Gamma}_1,\ldots,\boldsymbol{\Gamma}_n\},\\ &\boldsymbol{S} = diag\{\boldsymbol{S}_1,\ldots,\boldsymbol{S}_n\}, && \boldsymbol{T} = diag\{\boldsymbol{T}_1,\ldots,\boldsymbol{T}_n\},\\ &\boldsymbol{\Xi} = diag\{\overline{\boldsymbol{\Gamma}}_1\overline{\boldsymbol{K}}_1,\ldots,\overline{\boldsymbol{\Gamma}}_n\overline{\boldsymbol{K}}_n\}.\end{aligned} \tag{D.10}$$

By definitions (D.5) and (D.10), and substituting (D.3) and (D.4) into (D.9), the completed stacked open-loop optimal sequence can be expressed as

$$\begin{aligned}\mathbb{U}(k,M|k) &= \boldsymbol{\Gamma}'\boldsymbol{\Gamma}\mathbb{U}(k-1,M|k-1)+\boldsymbol{\Xi}\{\boldsymbol{Y}^d(k+1,P|k)\\ &\quad-\boldsymbol{S}[\overline{\boldsymbol{B}}\boldsymbol{\Gamma}'\boldsymbol{\Gamma}\mathbb{U}(k-1,M|k-1)+\overline{\boldsymbol{A}}\hat{x}(k|k)+\tilde{\boldsymbol{A}}\boldsymbol{\Omega}\hat{\mathbb{X}}(k,P|k-1)\\ &\quad+\tilde{\boldsymbol{B}}\boldsymbol{\Pi}\mathbb{U}(k-1,M|k-1)]-\boldsymbol{T}\tilde{\boldsymbol{C}}\boldsymbol{\Omega}\hat{\mathbb{X}}(k,P|k-1)\}\end{aligned} \tag{D.11}$$

Define

$$\begin{aligned}&\boldsymbol{\Theta} = -\boldsymbol{\Xi}\boldsymbol{S}\overline{\boldsymbol{A}}\\ &\boldsymbol{\Phi} = -\boldsymbol{\Xi}(\boldsymbol{S}\tilde{\boldsymbol{A}}\boldsymbol{\Omega}+\boldsymbol{T}\tilde{\boldsymbol{C}}\boldsymbol{\Omega}),\\ &\boldsymbol{\Psi} = \boldsymbol{\Gamma}'\boldsymbol{\Gamma}-\boldsymbol{\Xi}\boldsymbol{S}(\overline{\boldsymbol{B}}\boldsymbol{\Gamma}'\boldsymbol{\Gamma}+\tilde{\boldsymbol{B}}\boldsymbol{\Pi})\end{aligned} \tag{D.12}$$

Then the completed stacked open-loop optimal sequence (D.11) has the form

$$\mathbb{U}(k,M|k) = \boldsymbol{\Psi}\mathbb{U}(k-1,M|k-1)+\boldsymbol{\Theta}\hat{x}(k|k)+\boldsymbol{\Phi}\hat{\mathbb{X}}(k,P|k-1)+\boldsymbol{\Xi}\boldsymbol{Y}^d(k+1,P|k) \tag{D.13}$$

Therefore, the complete feedback control law computed by all controllers can be expressed as

$$\boldsymbol{u}(k) = \boldsymbol{\Gamma}\mathbb{U}(k,M|k) \tag{D.14}$$

Merging the process model (2), the feedback control law (D.14), the global prediction equation given by (36) and the controller equation given by (D.13), the closed-loop state-space representation for the distributed case is derived

$$\left\{\begin{array}{l}\boldsymbol{x}(k) = \boldsymbol{A}\boldsymbol{x}(k-1)+\boldsymbol{B}\boldsymbol{\Gamma}\mathbb{U}(k-1,M|k-1)\\ \hat{\mathbb{X}}(k,P|k-1) = \boldsymbol{L}\overline{\boldsymbol{S}}[\overline{\boldsymbol{A}}\hat{x}(k-1)+\tilde{\boldsymbol{A}}\boldsymbol{\Omega}\hat{\mathbb{X}}(k-1,P|k-2)+\overline{\boldsymbol{B}}\mathbb{U}(k-1,M|k-1)\\ \qquad\qquad+\tilde{\boldsymbol{B}}\boldsymbol{\Pi}\mathbb{U}(k-2,M|k-2)]\\ \mathbb{U}(k,M|k) = \boldsymbol{\Theta}\hat{x}(k)+\boldsymbol{\Phi}\hat{\mathbb{X}}(k,P|k-1)+\boldsymbol{\Psi}\mathbb{U}(k-1,M|k-1)\\ \qquad\qquad+\boldsymbol{\Xi}\boldsymbol{Y}^d(k+1,P|k)\\ \qquad\quad = \boldsymbol{\Theta}[\boldsymbol{A}\boldsymbol{x}(k-1)+\boldsymbol{B}\boldsymbol{\Gamma}\mathbb{U}(k-1,M|k-1)]\\ \qquad\qquad+\boldsymbol{\Phi}\boldsymbol{L}\overline{\boldsymbol{S}}[\overline{\boldsymbol{A}}\hat{x}(k-1)+\tilde{\boldsymbol{A}}\boldsymbol{\Omega}\hat{\mathbb{X}}(k-1,P|k-2)\\ \qquad\qquad+\overline{\boldsymbol{B}}\mathbb{U}(k-1,M|k-1)+\tilde{\boldsymbol{B}}\boldsymbol{\Pi}\mathbb{U}(k-2,M|k-2)]\\ \qquad\qquad+\boldsymbol{\Psi}\mathbb{U}(k-1,M|k-1)+\boldsymbol{\Xi}\boldsymbol{Y}^d(k+1,P|k)\\ \boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k)\end{array}\right. \tag{D.15}$$

where $\hat{x}(k|k)$ in Eqs. (D.8) and (D.13) has been substituted with $x(k)$ due to the assumption of fully accessible state. Defining the extended state $\boldsymbol{X}_N(k)=[\boldsymbol{x}^T(k)\ \ \hat{\mathbb{X}}^T(k,P|k-1)\ \ \mathbb{U}^T(k,M|k)\ \ \mathbb{U}^T(k-1,M|k-1)]^T$, the closed-loop state-space representation has the form

$$\left\{\begin{array}{l}\boldsymbol{X}_N(k) = \boldsymbol{A}_N\boldsymbol{X}_N(k-1)+\boldsymbol{B}_N\boldsymbol{Y}^d(k+1,p|k)\\ \boldsymbol{y}(k) = \boldsymbol{C}_N\boldsymbol{X}_N(k)\end{array}\right. \tag{D.16}$$

where

$$\boldsymbol{A}_N = \left[\begin{array}{cccc}\boldsymbol{A} & \boldsymbol{0} & \boldsymbol{B}\boldsymbol{\Gamma} & \boldsymbol{0}\\ \boldsymbol{L}\overline{\boldsymbol{S}}\overline{\boldsymbol{A}} & \boldsymbol{L}\overline{\boldsymbol{S}}\tilde{\boldsymbol{A}}\boldsymbol{\Omega} & \boldsymbol{L}\overline{\boldsymbol{S}}\overline{\boldsymbol{B}} & \boldsymbol{L}\overline{\boldsymbol{S}}\tilde{\boldsymbol{B}}\boldsymbol{\Pi}\\ \boldsymbol{\Theta}\boldsymbol{A}+\boldsymbol{\Phi}\boldsymbol{L}\overline{\boldsymbol{S}}\overline{\boldsymbol{A}} & \boldsymbol{\Phi}\boldsymbol{L}\overline{\boldsymbol{S}}\tilde{\boldsymbol{A}}\boldsymbol{\Omega} & \boldsymbol{\Theta}\boldsymbol{B}\boldsymbol{\Gamma}+\boldsymbol{\Phi}\boldsymbol{L}\overline{\boldsymbol{S}}\overline{\boldsymbol{B}}+\boldsymbol{\Psi} & \boldsymbol{\Phi}\boldsymbol{L}\overline{\boldsymbol{S}}\tilde{\boldsymbol{B}}\boldsymbol{\Pi}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{Mn_u} & \boldsymbol{0}\end{array}\right] \tag{D.17}$$

Thus Theorm 2 is obtained.

References

Bakule, L. (2008). Decentralized control: An overview. *Annual Reviews in Control*, *32*(1), 87–98.

Camponogara, E., Jia, D., & Krogh, B. H., et al. (2002). Distributed model predictive control. *IEEE Control Systems Magazine*, *22*(1), 44–52.

Du, X., Xi, Y., & Li, S. (2001). Distributed model predictive control for large-scale systems. In: Proceedings of the American control conference 2001 (pp. 3142–3143). Vol. 4.

Dunbar, W. B. (2007). Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Transactions on Automatic Control*, *52*(7), 1249–1263.

Dunbar, W. B., & Murray, R. M. (2006). Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, *42*(4), 549–558.

Guan, S., Wang, X., & Chai, T. (1998). An expert controller for the laminar cooling process of hot rolled slab. In: *Proceedings of the American control conference 1998* (pp. 2165–2169). *Vol. 4*.

Jin, N., Zhou, S., & Chang, T.-S., et al. (2008). Identification of influential functional process variables for surface quality control in hot rolling processes. *IEEE Transactions on Automation Science and Engineering*, *5*(3), 557–562.

Katebi, M. R., & Johnson, M. A. (1997). Predictive control design for large-scale systems. *Automatica*, *33*(3), 421–425.

Keviczky, T., Borrelli, F., & Balas, G. J. (2006). Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, *42*(12), 2105–2115.

Lee, S., Kumara, S., & Gautam, N. (2008). Market-based model predictive control for large-scale information networks: Completion time and value of solution. *IEEE Transactions on Automation Science and Engineering*, *5*(4), 630–640.

Li, S., Zhang, Y., & Zhu, Q. (2005). Nash-optimization enhanced distributed model predictive control applied to the Shell benchmark problem. *Information Sciences*, *170*(2–4), 329–349.

Li, S., Zheng, Y., & Wang, B. (2008). Steady-state target calculation for constrained predictive control systems based on goal programming. *Asia-Pacific Journal of Chemical Engineering*, *3*(6), 648–655.

Maciejowski, J. M. (2002). *Predictive control with constraints*. London, U.K: Prentice-Hall.

Magni, L., & Scattolini, R. (2006). Stabilizing decentralized model predictive control of nonlinear systems. *Automatica*, *42*(7), 1231–1236.

Mercangoz, M., & Iii, F. J.Doyle (2007). Distributed model predictive control of an experimental four-tank system. *Journal of Process Control*, *17*(3), 297–308.

Author's personal copy

Mukhopadhyay, A., & Sikdar, S. (2005). Implementation of an on-line run-out table model in a hot strip mill. *Journal of Materials Processing Technology*, *169*(2), 164–172.

Niederlinski, A. (1971). A heuristic approach to the design of linear multivariable interacting control systems. *Automatica*, *7*(6), 691–701.

Peng, H., Nakano, K., & Shioya, H. (2007). Nonlinear predictive control using neural nets-based local linearization arx model-stability and industrial application. *IEEE Transactions on Control Systems Technology*, *15*(1), 130–143.

Qin, S. J., & Badgwell, T. A. (2000). An overview of nonlinear model predictive control applications. In F. Allgower, & A. Zheng (Eds.), *Nonlinear Model Predictive Control* (pp. 369–392). Berlin: Birkhauser.

Qin, S. J., & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, *11*(7), 733–764.

Rawlings, J. B., & Stewart, B. T. (2008). Coordinating multiple optimization-based controllers: New opportunities and challenges. *Journal of Process Control*, *18*(9), 839–845.

Richalet, J. (1993). Industrial applications of model based predictive control. *Automatica*, *29*(5), 1251–1274.

Richards, A., & How, J. P. (2007). Robust distributed model predictive control. *International Journal of Control*, *80*(9), 1517–1531.

Scattolini, R. (2009). Architectures for distributed and hierarchical Model Predictive Control—A review. *Journal of Process Control*, *19*(5), 723–731.

Tang, L., Wang, X., & Liu, J. (2008). Color-coating production scheduling for coils in inventory in steel industry. *IEEE Transactions on Automation Science and Engineering*, *5*(3), 544–549.

Vaccarini, M., Longhi, S., & Katebi, M. R. (2009). Unconstrained networked decentralized model predictive control. *Journal of Process Control*, *19*(2), 328–339.

Van Henten, E. J., & Bontsema, J. (2009). Time-scale decomposition of an optimal control problem in greenhouse climate management. *Control Engineering Practice*, *17*(1), 88–96.

Venkat, A. N., Hiskens, I. A., & Rawlings, J. B., et al. (2008). Distributed MPC strategies with application to power system automatic generation control. *IEEE Transactions on Control Systems Technology*, *16*(6), 1192–1206.

Venkat, A. N., Rawlings, J. B., & Wright S. J. (2007). Distributed model predictive control of large-scale systems. In: P*roceedings of the assessment and future directions of nonlinear model predictive control* (pp. 591–605). Berlin Heidelberg: Springer; 2007.

Wang, Z. J., Wu, Q. D., & Chai, T. Y. (2004). Optimal-setting control for complicated industrial processes and its application study. *Control Engineering Practice*, *12*(1), 65–74.

Xu, M., Li, S., & Cai, W. (2005). Cascade generalized predictive control strategy for boiler drum level. *ISA Transactions*, *44*(3), 399–411.

Zhang, Y., & Li, S. (2007). Networked model predictive control based on neighbourhood optimization for serially connected large-scale processes. *Journal of Process Control*, *17*(1), 37–50.

Zheng, Y., Li, S., & Wang, X. (2009). Distributed model predictive control for plant-wide hot-rolled strip laminar cooling process. *Journal of Process Control*, *19*(9), 1427–1437.

Zheng, Y., Li, S., & Wang, X. (2010). An approach to model building for accelerated cooling process using instance-based learning. *Expert Systems with Applications*, *37*(7), 5364–5371.

Zheng, Y., Li, S., & Wang, X. (2011). Horizon-varying model predictive control for accelerated and controlled cooling process. *IEEE Transactions on Industrial Electronics*, *58*(1), 329–336.

Zhou, P., Chai, T., & Wang, H. (2009). Intelligent optimal-setting control for grinding circuits of mineral processing process. *IEEE Transactions on Automation Science and Engineering*