Available online at www.sciencedirect.com

ScienceDirect

00

00 00 (2019) 1–27

Robust distributed model predictive control for systems of parallel structure within process networks

Shuzhan Zhang[a], Dongya Zhao[a*], Sarah K. Spurgeon[b]

a. Department of Chemical Equipment and Control Engineering, College of Chemical Engineering, China University of Petroleum (East China), Qingdao, China.

b. Department of Electronic and Electrical Engineering, University College London, Torrington Place, London, UK.

** Corresponding Author: dyzhao@upc.edu.cn*

Abstract

The parallel structure is one of the basic system architectures found in process networks. In order to achieve robust control of complex process networks, it is necessary to formulate control strategies that specifically accommodate the characteristics of such parallel systems. In this paper, the competitive coupling and competitive constraints in parallel systems are initially defined. A novel robust distributed model predictive control algorithm is then developed for such parallel systems which deals explicitly with competitive couplings, competitive constraints and uncertainties. The Lyapunov Method is used for the theoretical analysis which produces tractable linear matrix inequalities (LMI). Two simulation studies and an experimental trial are provided to validate the effectiveness of the proposed approach. These consider control of 40 user and 100 user gas boiler heating systems as well as control of two continuous stirred tank reactors (CSTRs) which are connected in parallel.

Keywords: Parallel system, competitive coupling, competitive constraint, robust distributed model predictive control, linear matrix inequality

1. Introduction

Modern industrial chemical plants consist of many process units arranged in a complex structure which produces a process network [1]. Such a process network will contain many subsystems interacting with each other through mass and energy interconnections [2]. Depending on the structure of the process interactions, process networks can be divided into two forms: series structure encompassing the multistage extraction process [3] and the multistage flow control system [4]; parallel structure, where examples are the process of ethanol production [5] and the gas boiler heating system [6]. Series structure is characterised by each subsystem being connected in series [7] so that the output of the former subsystem is the input of the latter subsystem [8, 9, 10]. In a parallel structure, all subsystems are connected on the same bus and the input and output of each subsystem is related to the other subsystems.

The series configuration has been extensively studied in the process control domain [11, 12]. In this case the input of each subsystem is only coupled with the immediately preceding subsystem [13]. There are many publications which consider the control of series systems [14]. Methods discussed include state feedback control [15], active disturbance rejection control [16] as well as model predictive control (MPC) [17]. MPC is widely recognized as a practical control approach to yield high performance for series systems [18, 19, 20], where all the information is gathered into one model predictive controller to effect global optimization and ensure high control performance [21].

However, with the development of modern industry, the size of the series system is increasing, resulting in a need to control ever greater numbers of parameters. A centralized control structure leads to the requirement for large-scale online computation which results in increasing requirements for dedicated, centralised computer infrastructure. The availability of lower cost computational devices has resulted in the development of distributed MPC (DMPC) for the control of series systems [22]. The distributed control paradigm is promising as simple controllers result which can communicate and cooperate with each another to achieve improved control performance and operational efficiency [23]. For the series system, DMPC solves the optimization problem by iterative computing and rolling optimization and can handle constraint problems flexibly [24]. There is a vast literature on control of series systems using DMPC [25], including work on distributed model predictive control based on domain optimization[26], distributed decoupling model predictive control [27], Nash optimization based DMPC [28] and dynamic coordination based DMPC [29].

In process networks, a parallel system covers that case where all subsystems are connected in parallel. The inputs are obtained from the same bus while the corresponding outputs converge to another bus. Each subsystem is competing with the other subsystems because of resource limitations and there are couplings between information, mass and energy among the subsystems. These couplings and constraints in the parallel system architecture are different to those in a series system. It is necessary to clearly describe the couplings and constraints within a parallel system before embarking on a control design.

The competitive couplings and constraints are initially defined to describe the characteristics of the parallel system. Assume that two subsystems, for example, are paralleled in the same bus. It is obvious that the sum of the inputs of the two subsystems must be less than or equal to the total input of the whole system. When the sum of the inputs of two subsystems is equal to the total input of the whole system (this is the most common case in practice [30]), if the input of one subsystem increases, the input to the other subsystem must reduce. If the outputs of each subsystem are also connected in parallel, the outputs have the same characteristics. The system will not achieve the control objective and may even become unstable if the couplings and constraints are not addressed appropriately. This kind of coupling is called competitive coupling, and this constraint is a key consideration of this paper.

The dynamic characteristics of parallel and series systems are distinct. The existing DMPC methods for series systems cannot be directly applied to parallel systems. The literature specifically on DMPC for parallel systems is not extensive. Many of the control problems relating to parallel systems are solved by general DMPC algorithms [31]. However, the competitive couplings and competitive constraints of parallel systems are special and have not been considered by general DMPC explicitly. Hence, the design of the general DMPC lacks pertinence to the characteristics of the parallel system. If the DMPC algorithm for general systems is applied to a parallel system directly, there will be considerable computation and even degraded control performance.

Modelling error cannot be avoided in practice and it is important to consider uncertainty within the parallel system architecture in process networks. Several robust control methodologies have been used in process networks to address problems of system uncertainty and disturbances [32, 33]. A robust control method offers an effective solution [34] and a number of researchers have explored control approaches for series systems, such as sensitivity-based robust state feedback control [35], robust control of linear discrete-time series systems with local and global constraints [36]. However, there is very little literature on parallel systems which are extensively used in practice.

The purpose of this paper is to develop a novel iterative online algorithm for RDMPC, which explicitly deals with competitive coupling, competitive constraints and uncertainties within a parallel system. A linear matrix inequality (LMI) based RDMPC algorithm has been developed. It will be shown that the proposed iterative algorithm can be formulated so as to guarantee stability and to achieve the desired control objectives. Two challenging issues are considered in this paper: (1) Define the competitive coupling and competitive constraint, find the balance between the various subsystems of the competition, solve the competitive problem within a parallel system; (2) By using the Lyapunov stability theory and LMIs, the explicit RDMPC algorithm is designed for parallel systems and a sufficient condition is given for the existence of the control law. After addressing these two issues, stability of the resulting closed-loop system is analyzed. A step by step control algorithm is given to illustrate the usability. Finally, the effectiveness and performance are validated by extensive simulation results and an experimental trial.

The paper is organized as follows. Section 2 sees the parallel system problem being formulated and assumptions and definitions are given. In Section 3 the RDMPC algorithm is proposed and its robust stability is addressed. The results of simulation testing and an experimental trial are demonstrated in Section 4 in order to validate the proposed approach. Finally, some conclusions are drawn in Section 5.

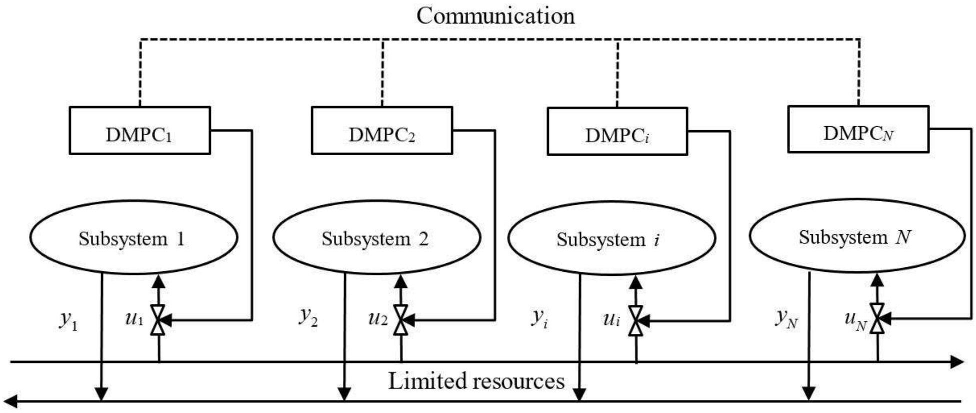

Figure 1: Parallel structure in process networks.

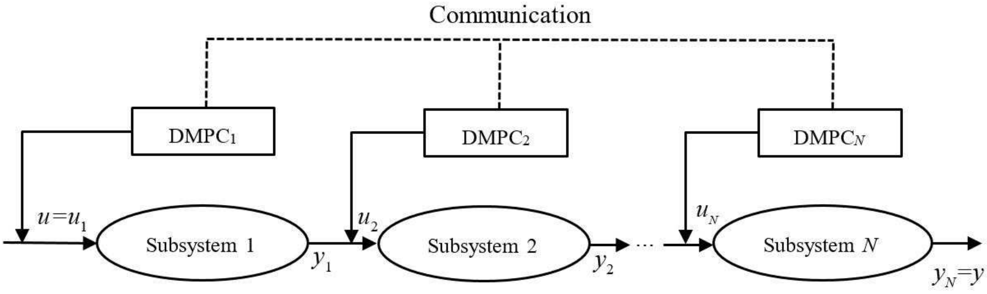

Figure 2: Series structure in process networks.

2. Problem formulation

The parallel and series structures in process networks are shown diagrammatically in Figures 1 and 2, respectively. The differences in the structural form between the two systems are clear. In the series system, the input of each subsystem is only coupled with the former subsystems [36] but with the parallel system each subsystem is coupled to all the other subsystems via the input and output. Considering uncertainties, the state equation of the *i*th subsystem of a parallel structure system with N subsystems can be written as:

$$\begin{aligned} x_i(k+1) &= (A_i + \Delta A_i)x_i(k) + (B_i + \Delta B_i)u_i(k) + \sum_{j=1, j\neq i}^{N} \lambda_{ji}{}^T (x_i(k) - \rho_{ji} x_j(k))(A_j + \Delta A_j)x_j(k)) \\ &\quad + \sum_{j=1, j\neq i}^{N} \delta_{ji}{}^T (u_i(k) - \sigma_{ji} u_j(k))(B_j + \Delta B_j)u_j(k) \\ y_i(k) &= x_i(k) \end{aligned} \tag{1}$$

where $x_i(k) \in \mathbb{R}^n$, $u_i(k) \in \mathbb{R}^r$ and $y_i(k) \in \mathbb{R}^n$ are respectively the state, input and output of the *i*th subsystem, $\sigma_{ji} \in \mathbb{R}^{r\times r}$, $\rho_{ji} \in \mathbb{R}^{n\times n}$, $i, j = 1, \cdots, N, j \neq i$ are weighting matrices representing the competitive strength of the input and output respectively, N denotes the number of subsystems. All elements of these matrices are positive. $\delta_{ji} \in \mathbb{R}^r$, $\lambda_{ji} \in \mathbb{R}^n$, $i, j = 1, \cdots, N, j \neq i$ are weighting vectors of the competitive coupling of the input and output respectively where all elements in the vectors are positive. It is assumed that the inputs and outputs are not directly related, and the observation matrix equals the identity matrix ($y_i(k) = x_i(k)$). $A_i \in \mathbb{R}^{n\times n}$, $B_i \in \mathbb{R}^{n\times r}$ are known constant matrices, $\Delta A_i \in \mathbb{R}^{n\times n}$, $\Delta B_i \in \mathbb{R}^{n\times r}$ are uncertainties of the ith subsystem and $\Delta A_j \in \mathbb{R}^{n\times n}$, $\Delta B_j \in \mathbb{R}^{n\times r}$ are uncertainties of the jth subsystem, with the following form:

$$\begin{bmatrix} \Delta A_i & \Delta B_i \end{bmatrix} = D_i F_i \begin{bmatrix} E_{i1} & E_{i2} \end{bmatrix}$$
$$\begin{bmatrix} \Delta A_j & \Delta B_j \end{bmatrix} = D_j F_j \begin{bmatrix} E_{j1} & E_{j2} \end{bmatrix} \tag{2}$$

where $F_i, F_j \in \mathbb{R}^{l\times m}$ are unknown real valued bounded functions and ${F_i}^T F_i \leq I_i$, ${F_j}^T F_j \leq I_j$, I_i, I_j are the identity matrices with appropriate dimensions, $D_i, D_j \in \mathbb{R}^{n\times l}$, $E_{i1}, E_{j1} \in \mathbb{R}^{m\times n}$, $E_{i2}, E_{j2} \in \mathbb{R}^{m\times r}$ are known constant matrices.

Definition 1. *For the ith subsystem of a parallel system, define*

$$c_i^I(k) = \sum_{j=1, j\neq i}^{N} {\delta_{ji}}^T (u_i(k) - \sigma_{ji} u_j(k))(B_j + \Delta B_j) u_j(k) \tag{3}$$

and

$$c_i^O(k) = \sum_{j=1, j\neq i}^{N} {\lambda_{ji}}^T (x_i(k) - \rho_{ji} x_j(k))(A_j + \Delta A_j) x_j(k) \tag{4}$$

as the competitive couplings.

In (3) and (4), $c_i^I(k)$ and $c_i^O(k)$ denote the competitive couplings of the input and output, respectively.

Definition 2. *The system with competitive coupling must satisfy $\sum \|u_i(k)\| \leq \|u(k)\|$, ${\delta_{ji}}^T (u_i(k) - \sigma_{ji} u_j(k)) > 0$ and ${\lambda_{ji}}^T (x_i(k) - \rho_{ji} x_j(k)) > 0$, where $u(k)$ is the total input, $i, j = 1, \cdots, N$, $j \neq i$. These constraints are called competitive constraints.*

Remark 1. *In the system (1), the detailed couplings are given in Definition 1. When $u_j(k)$ in a parallel system increases, $u_i(k)$ has to reduce correspondingly because of the limited total input. The degree of reduction of the input $u_i(k)$ is affected by the weighting of the competitive strength σ_{ji} and the weighting of the competitive coupling δ_{ji}. As for the output, the output of each subsystem is competing and when $y_j(k)$ increases, $y_i(k)$ must reduce.*

Let $R_j = {\delta_{ji}}^T (u_i(k) - \sigma_{ji} u_j(k))$, $Q_j = {\lambda_{ji}}^T (x_i(k) - \rho_{ji} x_j(k))$, which are the weight coefficients. Note that, in this paper $x_i(k)$ and $x_j(k)$ are assumed measurable. In the calculation, $x_i(k)$ and $x_j(k)$, $u_i(k)$ and $u_j(k)$ can be obtained. δ_{ji}, σ_{ji} in R_j and λ_{ji}, ρ_{ji} in Q_j are constants that can be modeled. Thus R_j and Q_j can be calculated at time k. Then, the model of the ith subsystem can be rewritten as:

$$\begin{aligned} x_i(k+1) &= (A_i + \Delta A_i) x_i(k) + (B_i + \Delta B_i) u_i(k) + \sum_{j=1, j\neq i}^{N} Q_j (A_j + \Delta A_j) x_j(k)) \\ &\quad + \sum_{j=1, j\neq i}^{N} R_j (B_j + \Delta B_j) u_j(k) \\ y_i(k) &= x_i(k) \end{aligned} \tag{5}$$

In (1), all elements are significant. Note that the general DMPC algorithm can be used to control parallel systems. However, if the competitive couplings and competitive constraints are not considered appropriately in the controller design, the control performance cannot be guaranteed.

For (5), the structure of the state feedback control is:

$$u_i(k+\tau|k) = K_i x_i(k+\tau|k) \tag{6}$$

where $K_i \in \mathbb{R}^{r\times n}$ is the state feedback gain matrix, $k > 0$, $\tau > 0$ is the τ^{th} predicted step. $u_i(k+\tau|k)$ is the predictive control at $k+\tau$ computed by the optimization problem at time k.

Considering the competitive constraints, the performance index [18] is:

$$\min_{K_i} \max_{K_i(k+\tau|k)} J_i(k) \tag{7}$$

such that

$$\begin{aligned} u_{i,\min} &\leq u_i(k) \leq u_{i,max} \\ \sum \|u_i(k)\| &\leq \|u(k)\| \\ \delta_{ji}{}^T(u_i(k) - \sigma_{ji}u_j(k)) &> 0 \\ \lambda_{ji}{}^T(x_i(k) - \rho_{ji}x_j(k)) &> 0 \end{aligned} \tag{8}$$

where

$$J_i(k) = \sum_{k=0}^{\infty} \left\{ x_i^T(k)H_{i1}x_i(k) + u_i^T(k)H_{i2}u_i(k) + \sum_{j=1,j\neq i}^{N} \left[x_j^T(k)H_{j1}x_j(k) + u_j^T(k)H_{j2}u_j(k) \right] \right\} \tag{9}$$

where $H_{i1} \in \mathbb{R}^{n\times n}$, $H_{i2} \in \mathbb{R}^{r\times r}$, $H_{j1} \in \mathbb{R}^{n\times n}$, $H_{j2} \in \mathbb{R}^{r\times r}$ are weighted symmetric positive definite matrices. $\sum_{j=1,j\neq i}^{N} x_j^T(k)$ $H_{j1}x_j(k)$ is the influence of the jth subsystem's output on the ith subsystem, while $\sum_{j=1,j\neq i}^{N} u_j^T(k)H_{j2}u_j(k)$ corresponds to that of the input.

The following assumptions are given:

Assumption 1. *Each (A_i, B_i) is stabilizable and x_i is measurable.*

Assumption 2. *The controllers are synchronous.*

Assumption 3. *The controllers communicate only once within a sampling time interval.*

These assumptions are not restrictive. Assumption 1 is a common requirement. Assumption 2 is not unduly strong because in process control systems the sampling interval is typically long when compared to the computational time. Assumption 3 is appropriate because single information exchange within a sampling time interval is consistent with minimizing the amount of data exchange via the network. The sampling time is k, $x_i(k) = x_i(k|k)$ denotes the value of the state measurement at the kth sampling instant [37]. At the time k, $u_i(k)$ can be obtained, which shows the control input sequence can be obtained according to the performance index (9). At the time period $[k, (k+1)]$, the closed loop equation is given by:

$$x_i(k+1) = (A_i + \Delta A_i + B_iK_i + \Delta B_iK_i)x_i(k) + \sum_{j=1,j\neq i}^{N} [Q_j(A_j + \Delta A_j) + R_j(B_jK_j + \Delta B_jK_j)]x_j(k) \tag{10}$$

The control objective is to design a robust distributed model predictive state feedback controller for the parallel system with uncertainty (5) and to solve the optimization problem (7) in the presence of the competitive couplings (3), (4) and competitive constraints (8) in order to determine a robust distributed model predictive control law (6) at every sampling time so that the uncertain parallel system meets the feasibility condition and achieves asymptotic stability.

Remark 2. *Although $u_i(k)$ can be obtained by the proposed approach and general DMPC [38] at sampling time k, it is found that large computation times and slow convergence exist in the general DMPC implementation due to the Jacobian iteration in solving the optimal problem (7). Consequently, to speed up convergence in the optimization problem (7), the* **successive iteration** *method [26] is used in this paper. This is applied to the solution of (7) for each subsystem to calculate a weighted average value between the current and previous iterate solutions. The details will be presented and the examples will show that the robust DMPC proposed in this paper can reduce the computation time and speed up convergence for parallel system architectures.*

3. Robust distributed model predictive control for parallel systems

3.1. Robust DMPC algorithm

In order to develop the on-line algorithm to achieve robust DMPC for parallel systems, the optimization problem (7-9) must be transformed into a solvable problem. It is assumed that there exists a quadratic function $V(x_i(k))$ making $J(x_i(k)) \leq V(x_i(k))$ [18]. The infinite time domain "min-max" optimization problem is then transformed into a solvable finite time "min" optimization problem.

Consider the following quadratic Lyapunov function:

$$\begin{aligned} V(x_i(k)) &= x_i^T(k)P_{i1}x_i(k) + u_i^T(k)P_{i2}u_i(k) + \sum_{\xi=-d_1}^{0} x_i^T(k+\xi)R_{i1}x_i(k+\xi) + \sum_{\xi=-d_2}^{0} u_i^T(k+\xi)R_{i2}u_i(k+\xi) \\ &+ \sum_{j=1,j\neq i}^{N}\left[\sum_{\xi=-d_3}^{0} x_j^T(k+\xi)R_{j1}x_j(k+\xi) + \sum_{\xi=-d_4}^{0} u_j^T(k+\xi)R_{j2}u_j(k+\xi)\right] \end{aligned} \tag{11}$$

where $P_{i1} = P_{i1}{}^T \in \mathbb{R}^{n\times n}$, $P_{i2} = P_{i2}{}^T \in \mathbb{R}^{r\times r}$, $R_{i1} \in \mathbb{R}^{n\times n}$, $R_{i2} \in \mathbb{R}^{r\times r}$, $R_{j1} \in \mathbb{R}^{n\times n}$, $R_{j2} \in \mathbb{R}^{r\times r}$ and all of them are positive definite matrices. d_1, d_2, d_3 and d_4 are positive numbers, d_1 denotes the predictive length of $x_i(k)$, d_2 denotes the predictive length of $u_i(k)$, d_3 denotes the predictive length of $x_j(k)$ and d_4 denotes the predictive length of $u_j(k)$.

At each sampling time k, $x_i(k) = x_i(k\,|k)$, suppose $V(x_i(k+\tau\,|k))$ satisfies the following inequality for all $x_i(k+\tau\,|k)$, $u_i(k+\tau\,|k)$ [37]:

$$\Delta V(x_i(k+\tau\,|k)) \quad \leq \quad -[x_i{}^T(k+\tau\,|k)P_{i1}x_i(k+\tau\,|k) + u_i{}^T(k+\tau\,|k)P_{i2}u_i(k+\tau\,|k)] \tag{12}$$

To make the robust performance objective function bounded, it must have $V(x_i(\infty\,|k)) = 0$ as $x_i(\infty\,|k) = 0$ [37]. Consider $u_i(k) = K_ix_i(k)$ and $J(x_i(k)) \leq V(x_i(k))$:

$$\begin{aligned} &\sum_{k=0}^{\infty}\left\{x_i^T(k)(H_{i1} + K_i^T H_{i2}K_i)x_i(k) + \sum_{j=1,j\neq i}^{N}\left[x_j^T(k)(H_{j1} + K_j^T H_{j2}K_j)x_j(k)\right]\right\} \\ \leq\ & x_i^T(k)(P_{i1} + K_i^T P_{i2}K_i)x_i(k) \\ &+ \sum_{\xi=-d_1}^{0} x_i^T(k+\xi)R_{i1}x_i(k+\xi) + \sum_{\xi=-d_2}^{0} x_i^T(k+\xi)K_i^T R_{i2}K_ix_i(k+\xi) \\ &+ \sum_{j=1,j\neq i}^{N}\left[\sum_{\xi=-d_3}^{0} x_j^T(k+\xi)R_{j1}x_j(k+\xi) + \sum_{\xi=-d_4}^{0} x_j^T(k+\xi)K_j^T R_{j2}K_jx_j(k+\xi)\right] \end{aligned} \tag{13}$$

The optimization of performance index $J(x_i(k))$ can be transformed to the minimization of $\min_{K_i} V(x_i(k))$:

$$\begin{aligned} V(x_i(k)) &= x_i^T(k)P_{i1}x_i(k) + u_i^T(k)P_{i2}u_i(k) + \sum_{\xi=-d_1}^{0} x_i^T(k+\xi)R_{i1}x_i(k+\xi) + \sum_{\xi=-d_2}^{0} u_i^T(k+\xi)R_{i2}u_i(k+\xi) \\ &+ \sum_{j=1,j\neq i}^{N}\left[\sum_{\xi=-d_3}^{0} x_j^T(k+\xi)R_{j1}x_j(k+\xi) + \sum_{\xi=-d_4}^{0} u_j^T(k+\xi)R_{j2}u_j(k+\xi)\right] \end{aligned} \tag{14}$$

Thus, the optimization problem, $\min_{K_i}\ \max_{K_i(k+\tau|k)}\ J_i(k)$, is transformed into a solvable finite time "min" optimization problem (14). The explicit expression of the control law is given by Theorem 1.

Lemma 1. *[39] Assume that matrices $\bar{Y}$, $\bar{D}$ have appropriate dimension, and $\bar{Y}$ is a symmetric matrix, then*

$$\bar{Y} + \bar{D}\bar{F} + \bar{F}^T\bar{D}^T < 0$$

for all matrices under the condition $\bar{F}^T\bar{F} \leq I$, if and only if there is a constant $\varepsilon > 0$, $\bar{Y} + \varepsilon\bar{D}\bar{D}^T + \varepsilon^{-1} < 0$.

Theorem 1. *For (5), under Assumptions 1-3, the state feedback gain matrix K_i can be solved by minimizing (14) with constraints (17-18):*

$$K_i = (X_{i2}S_iX_{i2}^T + \sum_{j=1,j\neq i}^{N} X_{j2}S_jX_{j2}^T)^{-T} Y_i^T (X_{i1}W_iX_{i1}^T + \sum_{j=1,j\neq i}^{N} X_{j1}W_jX_{j1}^T)^{-T} \tag{15}$$

where $X_{i2}, X_{j2} \in \mathbb{R}^{r\times r}$, $S_i, S_j \in \mathbb{R}^{r\times r}$, $Y_i \in \mathbb{R}^{n\times r}$, $X_{i1}, X_{j1} \in \mathbb{R}^{n\times n}$, $W_i, W_j \in \mathbb{R}^{n\times n}$ can be obtained by solving the following optimization problem:

$$\min_{W_i,W_j,Y_i,S_i,S_j,X_{i1},X_{i2},X_{j1},X_{j2},M_{i1},M_{i2},M_{j1},M_{j2},\varepsilon_i,\gamma_i,\varphi_i} [\gamma_i + \varphi_i + tr(M_{i1}) + tr(M_{i2}) + tr(M_{j1}) + tr(M_{j2})] \tag{16}$$

such that

$$\begin{bmatrix} \gamma_i & x_i^T(k)X_{i1} \\ X_{i1}^T x_i(k) & W_i \end{bmatrix} \geq 0 \quad \begin{bmatrix} M_{i1} & N_{i1} \\ N_{i1}^T & X_{i1} \end{bmatrix} > 0 \quad \begin{bmatrix} M_{i2} & N_{i2} \\ N_{i2}^T & X_{i2} \end{bmatrix} > 0$$

$$\begin{bmatrix} \varphi_i & u_i^T(k)X_{i2} \\ X_{i2}^T u_i(k) & S_i \end{bmatrix} \geq 0 \quad \begin{bmatrix} M_{j1} & N_{j1} \\ N_{j1}^T & X_{j1} \end{bmatrix} > 0 \quad \begin{bmatrix} M_{j2} & N_{j2} \\ N_{j2}^T & X_{j2} \end{bmatrix} > 0 \tag{17}$$

$$\begin{bmatrix} A^M & B^M \\ * & C^M \end{bmatrix} < 0 \tag{18}$$

$$A^M = \begin{bmatrix} L_i & 0 & 0 & (E_{i1}Z_i + E_{i2}Y_i)^T \\ * & -X_{i1} & 0 & 0 \\ * & * & -X_{i2} & 0 \\ * & * & * & * \end{bmatrix}$$

$$B^M = \begin{bmatrix} Y_i^T & Y_i^T & Z_i & Z_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad C^M = \begin{bmatrix} -R_i^{-1} & 0 & 0 & 0 \\ * & -X_{j2} & 0 & 0 \\ * & * & -X_{j1} & 0 \\ * & * & * & -P_i^{-1} \end{bmatrix}$$

$$L_i = A_iZ_i^T + B_iY_i^T + (A_iZ_i^T + B_iY_i^T)^T + \varepsilon_i D_i D_i^T$$

$$Z_i = X_{i1}W_iX_{i1}^T + \sum_{j=1,j\neq i}^{N} X_{j1}W_jX_{j1}^T$$

$$N_{i1}N_{i1}^T = \sum_{\xi=-d_1}^{0} x_i^T(k+\xi|k)x_i(k+\xi|k)$$

$$N_{i2}N_{i2}^T = \sum_{\xi=-d_2}^{0} u_i^T(k+\xi|k)u_i(k+\xi|k)$$

$$N_{j1}N_{j1}^T = \sum_{\xi=-d_3}^{0} x_j^T(k+\xi|k)x_j(k+\xi|k)$$

$$N_{j2}N_{j2}^T = \sum_{\xi=-d_4}^{0} u_j^T(k+\xi|k)u_j(k+\xi|k)$$

where $*$ indicates a symmetrical matrix element. M_{i1}, M_{i2}, M_{j1}, M_{j2}, γ_i, ε_i and φ_i will be defined in the following proof.

Proof:

Consider the following Lyapunov function at time k:

$$\begin{aligned} V(x_i(k)) &= x_i^T(k)P_{i1}x_i(k) + u_i^T(k)P_{i2}u_i(k) + \sum_{\xi=-d_1}^{0} x_i^T(k+\xi)R_{i1}x_i(k+\xi) + \sum_{\xi=-d_2}^{0} u_i^T(k+\xi)R_{i2}u_i(k+\xi) \\ &+ \sum_{j=1, j\neq i}^{N}\left[\sum_{\xi=-d_3}^{0} x_j^T(k+\xi)R_{j1}x_j(k+\xi) + \sum_{\xi=-d_4}^{0} u_j^T(k+\xi)R_{j2}u_j(k+\xi)\right] \end{aligned} \tag{19}$$

Assume that the parameter γ_i satisfies $x_i{}^T(k)P_{i1}x_i(k) \le \gamma_i$. According to the Schur complement lemma, $x_i{}^T(k)P_{i1}x_i(k) \le \gamma_i$ is equivalent to:

$$\begin{bmatrix} \gamma_i & x_i{}^T(k)X_{i1} \\ X_{i1}{}^T x_i(k) & W_i \end{bmatrix} \ge 0 \tag{20}$$

where $W_i = P_{i1}{}^{-T} \in \mathbb{R}^{n\times n}$, $X_{i1} = R_{i1}{}^{-1} \in \mathbb{R}^{n\times n}$.

Assume that the parameter φ_i satisfies $u_i{}^T(k)P_{i2}u_i(k) \le \varphi_i$. According to the Schur complement lemma, $u_i{}^T(k)P_{i2}u_i(k) \le \gamma_i$ is equivalent to:

$$\begin{bmatrix} \varphi_i & u_i{}^T(k)X_{i2} \\ X_{i2}{}^T u_i(k) & S_i \end{bmatrix} \ge 0 \tag{21}$$

where $S_i = P_{i2}{}^{-T} \in \mathbb{R}^{r\times r}$, $X_{i2} = R_{i2}{}^{-1} \in \mathbb{R}^{r\times r}$.

The first cumulative item in (19) can be converted into:

$$\begin{aligned} \sum_{\xi=-d_1}^{0} x_i{}^T(k+\xi|k)R_{i1}x_i(k+\xi|k) &= \sum_{\xi=-d_1}^{0} tr(x_i{}^T(k+\xi|k)X_{i1}{}^{-1}x_i(k+\xi|k)) \\ &= tr(N_{i1}N_{i1}{}^T X_{i1}{}^{-1}) = tr(N_{i1}{}^T X_{i1}{}^{-1}N_{i1}) \end{aligned} \tag{22}$$

where $X_{i1} = R_{i1}{}^{-1}$ and N_{i1} is as defined above and $N_{i1}N_{i1}{}^T = \sum_{\xi=-d_1}^{0} x_i{}^T(k+\xi|k)x_i(k+\xi|k)$. Assume that there is a matrix variable M_{i1} which satisfies $tr(N_{i1}{}^T X_{i1}{}^{-1}N_{i1}) < tr(M_{i1})$. According to the Schur complement lemma:

$$\begin{bmatrix} M_{i1} & N_{i1} \\ N_{i1}{}^T & X_{i1} \end{bmatrix} > 0 \tag{23}$$

In the same way, the second cumulative item in (19) can be converted into:

$$\begin{aligned} \sum_{\xi=-d_2}^{0} u_i{}^T(k+\xi|k)R_{i2}u_i(k+\xi|k) &= \sum_{\xi=-d_2}^{0} tr(u_i{}^T(k+\xi|k)X_{i2}{}^{-1}u_i(k+\xi|k)) \\ &= tr(N_{i2}K_iK_i{}^T N_{i2}{}^T X_{i2}{}^{-1}) = tr(K_i{}^T N_{i2}{}^T X_{i2}{}^{-1}N_{i2}K_i) \end{aligned} \tag{24}$$

where $X_{i2} = R_{i2}{}^{-1}$, N_{i2} is as defined above and $N_{i2}N_{i2}{}^T = \sum_{\xi=-d_2}^{0} u_i{}^T(k+\xi|k)u_i(k+\xi|k)$. Assume that there is a matrix variable M_{i2} which satisfies $tr(K_i{}^T N_{i2}{}^T X_{i2}{}^{-1}N_{i2}K_i) < tr(M_{i2})$. According to the Schur complement lemma:

$$\begin{bmatrix} M_{i2} & N_{i2} \\ N_{i2}{}^T & X_{i2} \end{bmatrix} > 0 \tag{25}$$

The last two cumulative items in (19) can be converted into:

$$\begin{aligned}
&\sum_{\xi=-d_3}^{0}\Big(\sum_{j=1,j\neq i}^{N} x_j{}^T(k+\xi|k)R_{j1}x_j(k+\xi|k)\Big) + \sum_{\xi=-d_4}^{0}\Big(\sum_{j=1,j\neq i}^{N} u_j{}^T(k+\xi|k)R_{j2}u_j(k+\xi|k)\Big)\\
&= \sum_{\xi=-d_3}^{0}\Big(tr\Big(\sum_{j=1,j\neq i}^{N} x_j{}^T(k+\xi|k)X_{j1}{}^{-1}x_j(k+\xi|k)\Big) + \sum_{\xi=-d_4}^{0}\Big(tr\Big(\sum_{j=1,j\neq i}^{N} u_j{}^T(k+\xi|k)X_{j2}{}^{-1}u_j(k+\xi|k)\Big)\\
&= tr\Big(\sum_{j=1,j\neq i}^{N} N_{j1}N_{j1}{}^T X_{j1}{}^{-1}\Big) + tr\Big(\sum_{j=1,j\neq i}^{N} N_{j2}K_jK_j{}^T N_{j2}{}^T X_{j2}{}^{-1}\Big)\\
&= tr\Big(\sum_{j=1,j\neq i}^{N} N_{j1}{}^T X_{j1}{}^{-1}N_{j1}\Big) + tr\Big(\sum_{j=1,j\neq i}^{N} K_j{}^T N_{j2}{}^T X_{j2}{}^{-1}N_{j2}K_j\Big)
\end{aligned}\tag{26}$$

where $X_{j1} = R_{j1}{}^{-1}$, $X_{j2} = R_{j2}{}^{-1}$. $N_{j1}N_{j1}{}^T = \sum_{\xi=-d_3}^{0} x_j{}^T(k+\xi|k)x_j(k+\xi|k)$, $N_{j2}N_{j2}{}^T = \sum_{\xi=-d_4}^{0} u_j{}^T(k+\xi|k)u_j(k+\xi|k)$. Assume that there are matrix variables M_{j1} and M_{j2}:

$$tr\Big(\sum_{j=1,j\neq i}^{N} N_{j1}{}^T X_{j1}{}^{-1}N_{j1}\Big) < tr(M_{j1})$$

$$tr\Big(\sum_{j=1,j\neq i}^{N} K_j{}^T N_{j2}{}^T X_{j2}{}^{-1}N_{j2}K_j\Big) < tr(M_{j2})$$

According to the Schur complement lemma:

$$\begin{bmatrix} M_{j1} & N_{j1}\\ N_{j1}{}^T & X_{j1}\end{bmatrix} > 0 \tag{27}$$

$$\begin{bmatrix} M_{j2} & N_{j2}\\ N_{j2}{}^T & X_{j2}\end{bmatrix} > 0 \tag{28}$$

Substitute (6) into (12):

$$\Delta V(x_i(k+\tau)) \leq -[x_i^T(k+\tau)P_{i1}x_i(k+\tau) + u_i^T(k+\tau)P_{i2}u_i(k+\tau)] = -[x_i^T(k+\tau)(P_{i1} + K_i^T P_{i2}K_i)x_i(k+\tau)] \tag{29}$$

Define:

$$\Omega_i = \begin{bmatrix} \Omega_{i1} & 0 & 0\\ 0 & -R_{i1} & 0\\ 0 & 0 & -R_{i2}\end{bmatrix}$$

where

$$\Omega_{i1} = (A_i + B_iK_i)^T P_{i1} + P_{i1}{}^T(A_i + B_iK_i) + R_{i1} + K_i{}^T R_{i2}K_i + P_{i1} + K_i{}^T P_{i2}K_i$$

Then, (29) is equivalent to:

$$\Omega_i + \begin{bmatrix} P_{i1}{}^T D_i\\ 0\\ 0\end{bmatrix} F_i \begin{bmatrix} E_{i1}K_i & 0 & 0\end{bmatrix} + \begin{bmatrix} E_{i1}K_i & 0 & 0\end{bmatrix}^T F_i{}^T \begin{bmatrix} P_{i1}{}^T D_i\\ 0\\ 0\end{bmatrix}^T \leq 0 \tag{30}$$

where E_{i1}, E_{i2}, D_i and F_i are defined after (1): $F_i \in \mathbb{R}^{l\times m}$ is an unknown real valued bounded function and $F_i{}^T F_i \leq I_i$, $D_i \in \mathbb{R}^{n\times l}$, $E_{i1} \in \mathbb{R}^{m\times n}$, $E_{i2} \in \mathbb{R}^{m\times r}$ are known constant matrices. According to Lemma 1, there is an $\varepsilon_i > 0$, which transforms the inequality into the following form:

$$\Omega_i + \varepsilon_i \begin{bmatrix} P_{i1}{}^T D_i\\ 0\\ 0\end{bmatrix}\begin{bmatrix} P_{i1}{}^T D_i\\ 0\\ 0\end{bmatrix}^T + \varepsilon_i{}^{-1}\begin{bmatrix} E_{i1} + E_{i2}K_i & 0 & 0\end{bmatrix}^T \begin{bmatrix} E_{i1} + E_{i2}K_i & 0 & 0\end{bmatrix} < 0 \tag{31}$$

Then, (31) can be transformed into a linear matrix inequality and it is known by the Schur complement lemma:

$$\begin{bmatrix} L_{i1} & 0 & 0 & (E_{i1}+E_{i2}K_i)^T \\ * & -R_{i1} & 0 & 0 \\ * & * & -R_{i2} & 0 \\ * & * & * & -\varepsilon_i I_i \end{bmatrix} < 0 \tag{32}$$

where:

$$L_{i1} = (A_i + B_iK_i)^T P_{i1} + P_{i1}{}^T (A_i + B_iK_i) + R_{i1} + K_i{}^T R_{i2} K_i + P_{i1} + K_i{}^T P_{i2} K_i + \varepsilon_i P_{i1}{}^T D_i D_i{}^T P_{i1}$$

Further more, according to the Schur complement lemma:

$$\begin{bmatrix} A^L & B^L \\ * & C^L \end{bmatrix} < 0 \tag{33}$$

where:

$$A^L = \begin{bmatrix} L_{i2} & 0 & 0 & (E_{i1}+E_{i2}K_i)^T \\ * & -R_{i1} & 0 & 0 \\ * & * & -R_{i2} & 0 \\ * & * & * & -\varepsilon_i I_i \end{bmatrix}$$

$$B^L = \begin{bmatrix} K_i{}^T & K_i{}^T & I_i & I_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C^L = \begin{bmatrix} -P_{i2}{}^{-1} & 0 & 0 & 0 \\ * & -R_{i2} & 0 & 0 \\ * & * & -R_{i1} & 0 \\ * & * & * & -P_{i1}{}^{-1} \end{bmatrix}$$

$$L_{i2} = (A_i + B_iK_i)^T P_{i1} + P_{i1}{}^T (A_i + B_iK_i) + \varepsilon_i P_{i1}{}^T D_i D_i{}^T P_{i1}$$

Multiplying the transformation matrix $diag\{Z_i, X_{i1}, X_{i2}, X_{j1}, X_{j2}, I_i, I_i, I_i, I_i\}$ on both sides of (33), then, inequality (33) is equivalent to inequality (18). The state feedback gain matrix K_i can be determined by minimizing (14). Q.E.D.

Remark 3. *Theorem 1 shows that the problem of robust DMPC for a parallel system can be transformed into a finite time "min" optimization problem under a set of linear matrix inequality constraints at each k instant. Inequality (18) is a linear matrix inequality that can be solved by the LMI toolbox. The procedure can be summarized in the following step by step algorithm.*

Algorithm 1.

Step 1 (Initialization): At control interval k=0, set K_i=0.

Step 2 (Updating): At the beginning of control interval (k) all the controllers exchange their state measurements and initial estimates K_i. Set iteration t=0 and K_i=$K_i{}^{(0)}$, where t is the iteration number.

Step 3 (Iterations): While $t \leq t_{\max}$, solve N LMI problems (18) in parallel to obtain $W_i{}^{(t)}$, $W_j{}^{(t)}$, $Y_i{}^{(t)}$, $S_i{}^{(t)}$, $S_j{}^{(t)}$, $X_{i1}{}^{(t)}$, $X_{i2}{}^{(t)}$, $X_{j1}{}^{(t)}$, $X_{j2}{}^{(t)}$, $M_{i1}{}^{(t)}$, $M_{i2}{}^{(t)}$, $M_{j1}{}^{(t)}$, $M_{j2}{}^{(t)}$, $\gamma_i{}^{(t)}$, $\varepsilon_i{}^{(t)}$, $\varphi_i{}^{(t)}$ to estimate the feedback solutions $K_i{}^{(t)}$. Check the convergence for a specified error tolerance e_i which is defined by users for all the controllers. If $K_i^{(t)}$ satisfies:

$$\left\| K_i{}^{(t)} - K_i{}^{(t-1)} \right\| \leq e_i, \forall i \in \{1, ..., N\}$$

then, go to step 4. Otherwise continue to iterate, exchange the solution K_i and set t=t+1.

Step 4 (Implementation): Apply the control $u_i = K_i x_i$ to the corresponding subsystems. Go to the control interval k=k+1, return to step 2 and repeat the procedure.

Remark 4. *Algorithm 1 computes state-feedback control laws for every subsystem and therefore full state measurement is assumed. In step 2, at control interval k, the feedback solutions can be obtained in the previous interval k-1 which uses the initial estimates $K_i^{(0)}$ to initialise the iteration. It should be mentioned that X_{i2}, M_{i2}, γ_i, $\varphi_i^{(t)}$ and ε_i are not considered in general DMPC. The method used in this study is* **successive iteration** *which can accelerate the convergence speed for the parallel system architecture. The successive iteration is given by the following equation: $K_i^{(t+1)} = \alpha\bar{K}_i^{(t+1)} + (1-\alpha)K_i^{(t)}$, where α is a filter parameter to be specified by the user. $\bar{K}_i^{(t+1)}$ denotes the solution obtained at the current iteration from (18) whereas $K_i^{(t+1)}$ is the estimate to be used in the next iteration. α can be chosen from values between 0 and 2. If it is chosen to be 1, the method corresponds to Jacobian iteration. Jacobian iteration is a special case of successive iteration. The stability of the proposed approach will be analyzed in the following subsection.*

Remark 5. *The prediction length $\mathbb{P}$ and control horizon $\mathbb{M}$ are the common parameters in DMPC which can influence the stability and computational complexity of the problem. The stability of the system is mildly affected by the control horizon $\mathbb{M}$ but heavily affected by the prediction length $\mathbb{P}$. If the selection of the prediction length $\mathbb{P}$ is too small, the stability and constraints may not be satisfied [9]. The computational complexity is influenced by these two parameters. If the value of $\mathbb{M}$ is larger, the number of control laws which are calculated in one iteration will be greater. If $\mathbb{P}$ is larger, then more information on the subsystems is needed to calculate the couplings. Larger $\mathbb{M}$ and $\mathbb{P}$ will produce computational complexity. Usually $\mathbb{M} \le \mathbb{P}$. In reference [18], the ranges of the control horizon $\mathbb{M}$ and prediction length $\mathbb{P}$ are given according to the sampling time T of the continuous system and dynamic response time of the system t_F: $\mathbb{M} + 1 \le \mathbb{P} \le \frac{2t_F}{T}$.*

3.2. Robust stability analysis

Lemma 2. *[40] Every feasible solution of optimization problem (18) in time k is still feasible in time $N(N > k)$.*

Theorem 2. *For (5), under the state feedback control K_i which is given by Theorem 1, if the optimization problem (14) has a feasible solution, then the closed loop system in (10) will be asymptotically stable.*

Proof:

According to Lemma 2, the optimization problem (7-9) is always feasible. Then from (29), it follows that

$$\Delta V(x_i(k|k)) \le -[x_i^T(k|k)P_{i1}x_i(k|k) + u_i^T(k)P_{i2}u_i(k|k)] = -[x_i^T(k|k)(P_{i1} + K_i^T P_{i2} K_i)x_i(k|k)] \tag{34}$$

P_{i1} and P_{i2} are weighted symmetric positive definite matrices, so that $\Delta V(x_i(k|k)) < 0$. The closed loop parallel system is asymptotically stable. Q.E.D.

4. Simulation and implementation results

The effectiveness of the proposed approach is validated by using two simulation studies and an experimental implementation. Firstly, a parallel gas boiler heating system is simulated. Two cases are considered corresponding to 40 users and 100 users, respectively. The users are paralleled in one heating net in each case and the problem of competitive coupling among the users is the problem of interest. Under the competitive constraints, the proposed approach is used to solve the heating demand for each user. The performance of the proposed approach is compared with general DMPC [33] and PI control. A simulation study and experimental implementation is then undertaken for a parallel continuous stirred tank reactor (CSTR) system. Matlab simulation is firstly undertaken and the performance is compared with general DMPC and PI control. Finally, a parallel CSTR experiment is used to further validate the proposed approach with comparison of the classical PID control.

4.1. Gas boiler heating system

A simplified physical model of a gas boiler heating system is given in Fig. 3. The medium of heat transfer is water. The water is heated through a heat source in the heat station before flowing out and eventually returning to the heat station. All users are connected to the network in parallel form. Hot water is controlled for each user by a valve. The hot water is limited. When users who are close to the heat station demand excessive hot water, it is difficult to

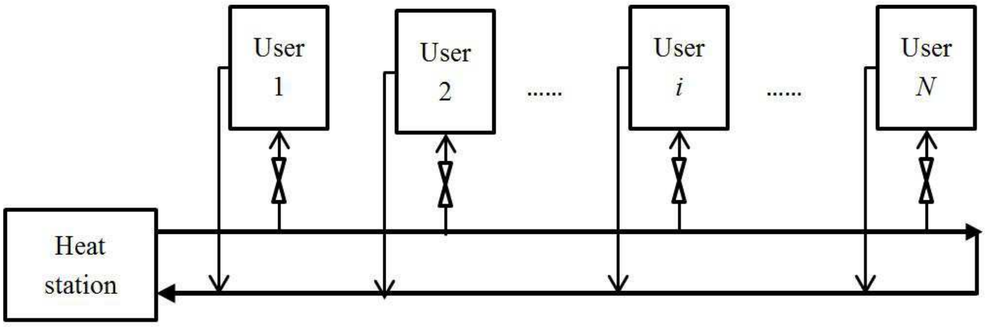

Figure 3: Simplified physical model of gas boiler heating system.

ensure the requirements of more distant users. The gas boiler heating system thus exhibits competitive coupling. The competitive constraint is that the total amount of hot water obtained by all users cannot exceed that provided by the heat station.

The backwater temperature of the second network is to be controlled where the transfer function is given by:

$$T_{outi}(s) = \frac{K_{ni}[T_{mi}s + (K_{mi} + 1)]}{T_{mi}s + 1} U_i(s) + \frac{1}{T_{mi}s + 1} T_i'(s) \tag{35}$$

where $K_{ni} = c_i \Delta T_i R_{ni}$, $K_{mi} = \frac{R_{mi}}{R_{ni}}$, $T_{mi} = R_{mi}C_i$, $R_{mi} = \frac{1}{r_i \bar{S}_i}$, $R_{ni} = \frac{1}{c\bar{Q}_i}$, $C = \bar{M}_i c_i^{'}$, T_{outi} is the temperature, $\bar{Q}_i$ is the flow in the heating network, c is the specific heat of water, $\bar{M}_i$ is the average mass of material inside a dwelling, $c_i^{'}$ is the average specific heat of the material inside dwelling i, r_i is the heat transfer coefficient of user i, $\bar{S}_i$ is the heat dissipation area, T_i' is the outdoor temperature, ΔT_i is the inlet water temperature minus the return water temperature in the heating network and i denotes the user i.

The control objective is to raise the water temperature for each user by 10 degrees Celsius by controlling the valve corresponding to each user under the competitive constraints. According to the collected data, the prediction model is established. The system uses the flow of the hot water as the manipulating variable, with two-way valve opening control, sampling period $\mathbb{T}_s = 5s$ and sampling length $\mathbb{N} = 100$. The predicted length $\mathbb{P} = 25$ and the control horizon is $\mathbb{M} = 1$, $\alpha = 0.95$, $d_1 = 3$, $d_2 = 4$, $d_3 = 3$ and $d_4 = 4$, $P_{11} = P_{21} = \cdots P_{i1} = \cdots P_{N1} = 1$, $P_{12} = P_{22} = \cdots P_{i2} = \cdots P_{N2} = 1$, $R_{11} = R_{21} = \cdots R_{i1} = \cdots R_{N1} = 1$, $R_{12} = R_{22} = \cdots R_{i2} = \cdots R_{N2} = 1$. The PI parameters are set according to the the ultimate-sensitivity method as in [41]: P (K_c) is 0.8, I (T_i) is 0.5. The desired temperature is 10 degrees Celsius.

In the first case, there are 40 subsystems ($N = 40$) to be controlled and simulation results are presented for the cases of a simple PI controller, robust DMPC and general DMPC. The results for the first two subsystems and the 34th subsystem are shown in Fig. 4, Fig. 5 and Fig. 6. The integral time multiplied square error (ITSE) results for 40 users are tabulated in Table 1.

It can be seen from Fig. 4, Fig. 5 and especially Fig. 6 that for the method proposed in this paper (solid line), three parallel users can achieve the desired temperature more rapidly than with the simple PI controller (dashed line) and general DMPC (dotted line). The control performance of the users are identical under the proposed approach. The computation time for the robust DMPC is 1.35 seconds while that of the general DMPC is 1.86 seconds. In Table 1, it can be seen that the ITSE results of the proposed method compare favourably with those obtained using PI control and general DMPC. In 88 % of cases, the proposed method provides better accuracy than general DMPC in terms of ITSE and it is better than the PI strategy in all cases. It should be noted that for users situated furthest from the heat station, the proposed robust DMPC yields consistently better performance. Recall that this level of performance is achieved with a shorter computation time.

To further validate the advantages of the proposed approach, the second case with 100 users ($N = 100$) is now presented and compared with the first case. The results for the first two subsystems and the 51st subsystem are shown in Fig. 7, Fig. 8 and Fig. 9.

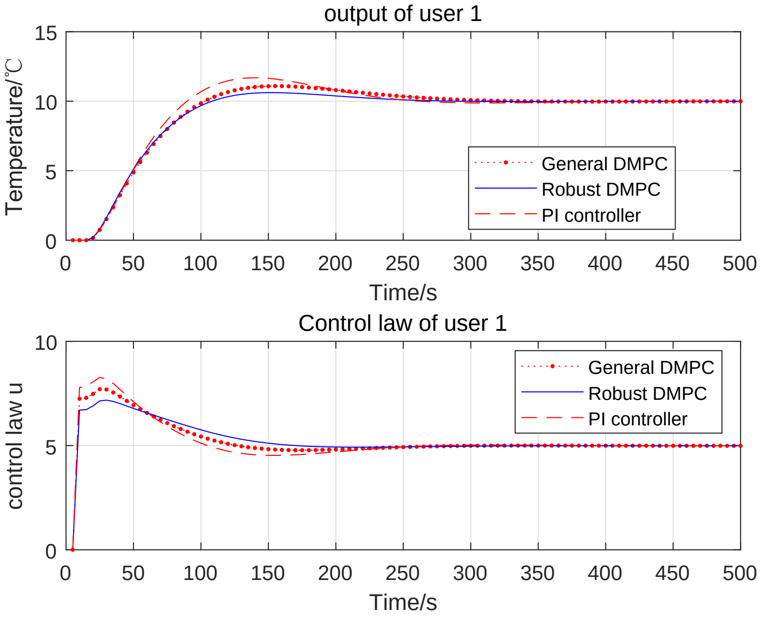

Figure 4: Results for user 1 (40 user scenario).

It can be seen from Fig. 7, Fig. 8 and Fig. 9 that for the method proposed in this paper (solid line), three parallel users achieve the desired temperature more rapidly than with the PI controller (dashed line) and general DMPC (dotted line). In comparison with Fig. 4 and Fig. 7, Fig. 5 and Fig. 8, it can be seen that the advantages of the proposed method are more pronounced with the increasing number of subsystems (from 40 to 100). The computation time for the robust DMPC with 100 subsystems is 2.17 seconds while that of the general DMPC is 5.35 seconds. In Table 1 corresponding to the case of 40 subsystems, 88% of subsystems have better performance with the proposed approach than with a general DMPC implementation. When the number of subsystems increases to 100, it can be seen from Table 2 that 94% of subsystems exhibit better performance with the proposed approach than with general DMPC. Comparing Table 1 and Table 2, it is again seen that when the number of subsystems increases, the advantages of the proposed approach become more pronounced.

4.2. Parallel CSTR system simulation

A simplified physical model of a parallel CSTR system is given in Fig. 10. The reaction is an endothermic reaction. If the temperature inside the reactor is low, it will affect the depth and conversion rate of the reaction, thus affecting the quality of the product. In order to stabilize the temperature inside the reactor, it is necessary to heat the jacket. The temperature of the material in the reactor can be controlled to meet the requirements of the process by adjusting the flow rate of the heat agent flowing into the jacket. The heat agent is water, which is supplied by one water heater. When the two CSTRs are connected in parallel, the system exhibits competitive coupling and there is a need to consider competitive constraints. When the flow of hot water in one jacket exceeds a certain amount, the flow of water in the second jacket must be be reduced. The heat agent flowing into the jacket must be less than or equal to the total supply of the water heater.

In this paper, the reaction temperature is taken as the controlled variable and the water flow in the jacket is the control variable. The control objective is to increase the temperature of the jacket from 0 degrees Celsius to 10 degrees Celsius by manipulating the flow of water in the jacket. The transfer function is given by:

$$G(s) = \frac{\tilde{T}(s)}{W(s)} = \frac{a_{12}b_{22}}{s^2 - (a_{11} + a_{22})s + (a_{11}a_{22} - a_{12}a_{21})} \tag{36}$$

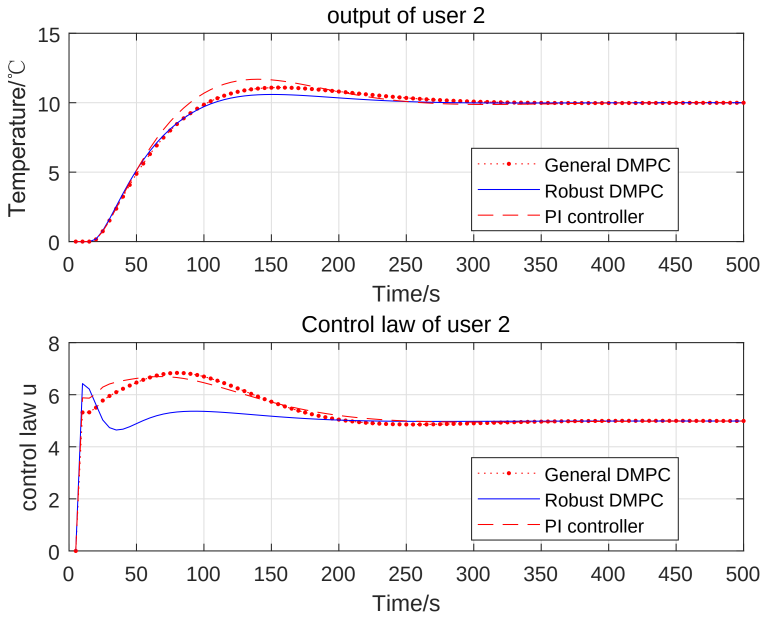

Figure 5: Results for user 2 (40 user scenario).

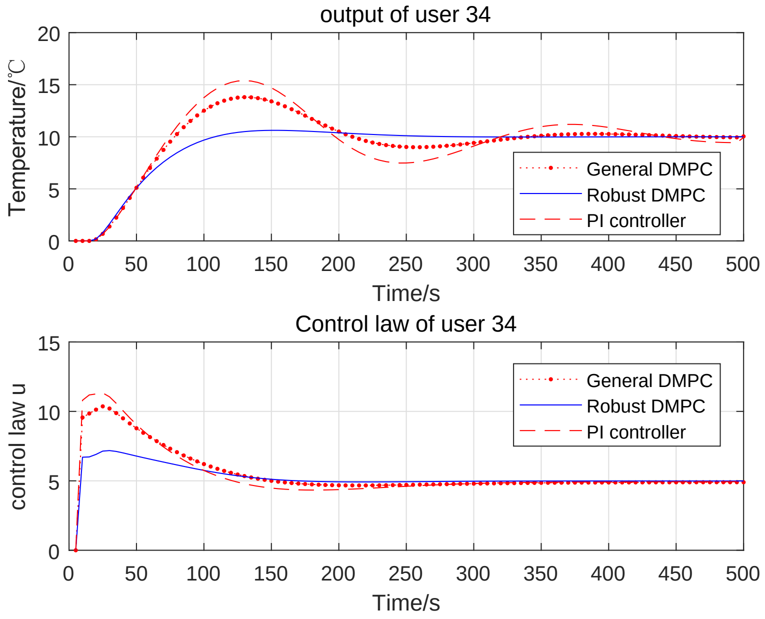

Figure 6: Results for user 34 (40 user scenario)

.

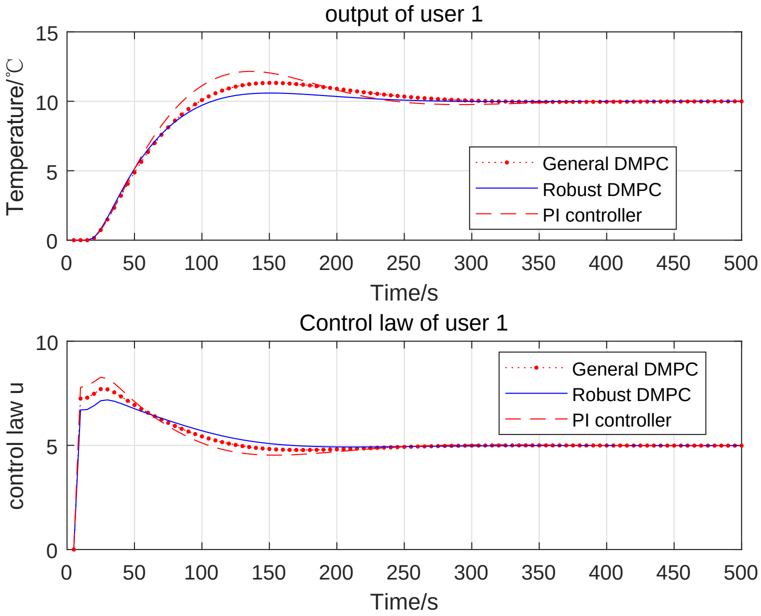

Figure 7: Results for user 1 (100 user scenario).

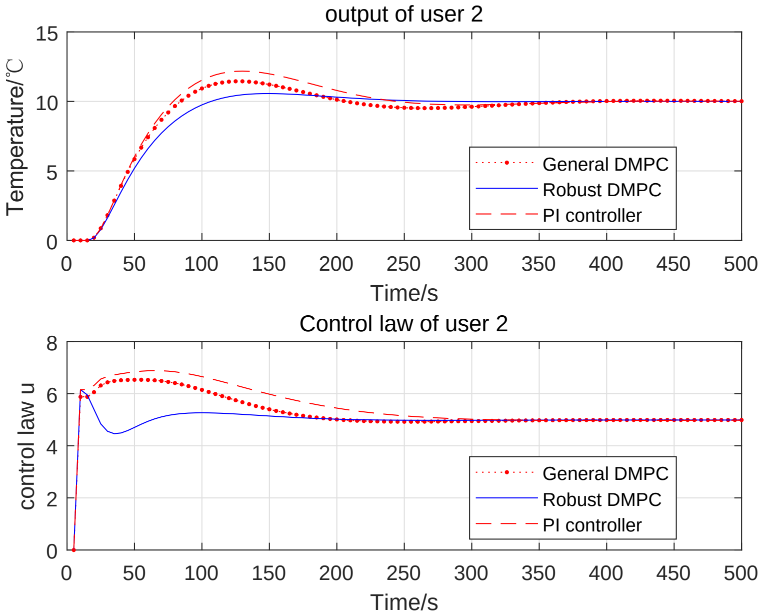

Figure 8: Results for user 2 (100 user scenario).

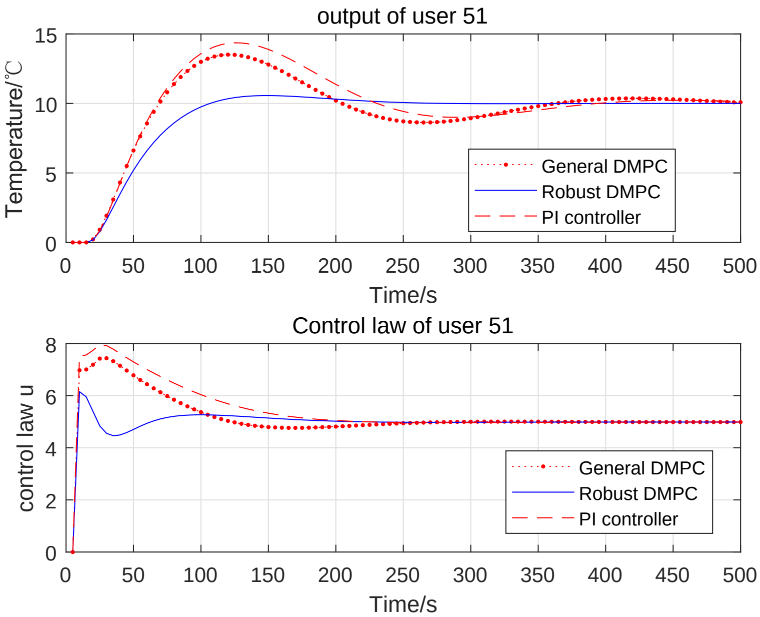

Figure 9: Results for user 51 (100 user scenario)

.

Table 1: ITSE results for 40 subsystem scenario

Subsystem	RDMPC	general DMPC	PI	Subsystem	RDMPC	general DMPC	PI
1	5.34815	5.49415	6.24011	21	5.33744	5.80374	5.98209
2	0.04057	0.05057	0.15457	22	0.00382	0.04138	0.08541
3	1.40301	1.40301	2.02474	23	1.35636	1.40661	1.98571
4	0.01728	0.11727	0.24041	24	0.01468	0.20848	0.31989
5	0.78574	0.43774	1.02485	25	2.10511	2.17754	3.02458
6	2.33454	2.33758	3.78524	26	5.22086	5.24237	5.40254
7	0.20453	0.20341	0.31578	27	1.40015	2.04671	3.05877
8	0.25603	0.23113	0.57587	28	0.04011	0.05394	0.12057
9	0.01861	0.04618	0.10523	29	1.45364	1.75858	1.97501
10	6.40327	6.20373	7.24564	30	2.11135	2.25858	3.02136
11	0.41968	0.43561	0.50127	31	5.32402	5.43529	6.01201
12	0.04057	0.04451	0.20157	32	0.04044	0.12541	0.11015
13	1.53184	1.54024	1.98545	33	1.33306	1.44847	1.98755
14	0.01864	0.14644	0.20578	34	0.01878	0.14281	0.09751
15	2.00105	2.01153	2.42204	35	2.09829	2.14914	3.02453
16	5.37153	5.43703	5.97125	36	4.30348	4.52082	5.02374
17	0.04032	0.07473	0.10578	37	0.04015	0.04791	0.09548
18	1.31063	1.36134	1.82410	38	1.31041	1.34161	1.53211
19	0.01773	0.12793	0.20054	39	0.01707	0.03982	0.08964
20	2.14175	2.14727	3.57409	40	2.10511	2.12753	2.56871

Table 2: ITSE results of 100 subsystem scenario

Subsystem	RDMPC	general DMPC	PI	Subsystem	RDMPC	general DMPC	PI
1	2.58462	3.22743	3.82548	51	0.25420	1.01455	1.45120
2	1.23875	1.84501	2.86541	52	3.20158	5.04564	4.21554
3	1.35714	1.86141	2.50564	53	2.01205	3.02787	3.65412
4	1.57821	2.62154	2.71401	54	1.02571	2.05327	2.59787
5	0.93238	1.20554	1.54014	55	2.01477	2.42063	2.78942
6	1.55128	2.05485	2.72014	56	0.56797	2.05367	1.98741
7	3.58841	5.21851	5.50244	57	1.02378	1.89731	1.98712
8	2.65248	3.57814	3.88789	58	2.00127	3.02102	3.47419
9	0.12852	0.53872	0.87410	59	1.01365	2.98977	2.91428
10	1.57521	2.87413	2.97211	60	5.20122	4.01289	5.97525
11	0.56711	0.98721	1.02458	61	1.28056	3.00278	3.12891
12	4.17414	4.02283	4.52216	62	2.01271	3.89701	3.97110
13	3.54871	5.20487	5.51237	63	0.20450	1.65051	1.97458
14	0.27541	0.85777	1.03278	64	1.02450	2.02768	2.15671
15	5.65871	6.28710	6.84201	65	1.02876	2.07026	2.72106
16	0.25784	1.45204	1.78506	66	2.04947	2.05678	2.69754
17	1.52785	2.07413	2.94104	67	1.02671	1.06977	1.21024
18	1.20473	1.33610	1.58997	68	2.06710	3.97021	3.45891
19	3.27411	2.98841	3.14204	69	1.06879	1.97015	2.01678
20	1.27485	2.56741	3.01569	70	1.05741	1.98701	2.03789
21	1.28987	1.98725	2.02378	71	3.03871	3.87073	4.02384
22	1.25841	1.89661	2.02137	72	1.03541	2.04576	2.30113
23	3.25741	4.49461	4.53201	73	2.01791	3.04578	4.02368
24	1.68410	2.87101	2.54210	74	1.02687	1.68740	2.03674
25	1.28779	2.57210	3.04176	75	0.89523	1.08954	2.02136
26	1.28987	3.51007	3.81405	76	2.03714	2.98037	2.88771
27	3.31289	4.52068	4.52001	77	1.25210	1.65001	1.97752
28	2.23698	2.98522	2.99522	78	0.94841	1.68921	1.98791
29	2.98413	4.02871	4.20215	79	5.66767	7.20545	6.98751
30	2.21187	3.20157	3.59870	80	1.05689	0.96840	1.98752
31	4.36841	6.21571	6.97452	81	2.02145	3.01227	3.65420
32	0.98125	1.85013	2.02378	82	1.05967	1.97207	2.02389
33	2.62101	2.58410	2.78717	83	2.04819	2.98043	3.46576
34	0.52507	1.85540	2.01253	84	0.58321	1.51078	1.93204
35	1.89320	2.54046	2.94105	85	0.09146	0.10289	0.23574
36	6.21871	5.98720	6.87142	86	6.04567	6.02568	6.65631
37	0.89504	1.52889	1.98712	87	2.03879	3.02876	3.68717
38	1.58964	2.63201	3.02379	88	0.97852	1.32161	3.97121
39	2.58710	3.15789	3.94019	89	0.79892	1.06780	2.02367
40	1.56843	2.39741	2.97454	90	1.96469	3.03254	3.61212
41	2.58241	3.49415	3.02567	91	3.24587	3.97542	4.67948
42	1.48734	2.15963	2.84506	92	0.97812	1.78020	1.98453
43	1.85404	1.95301	2.08871	93	3.15687	3.65202	4.02398
44	2.68417	2.97420	5.02458	94	1.20146	0.98014	1.69782
45	1.35853	3.20175	2.05887	95	1.20698	2.98701	2.42040
46	5.20105	6.69871	6.87013	96	3.02136	4.20554	4.89748
47	1.25836	1.96543	2.02375	97	0.68752	2.98752	3.01581
48	1.20447	2.15401	2.12076	98	1.57207	2.05867	2.65413
49	1.25847	2.67101	2.87142	99	2.51207	3.05869	3.08796
50	3.65588	5.01254	4.87811	100	0.87941	1.01691	1.35421

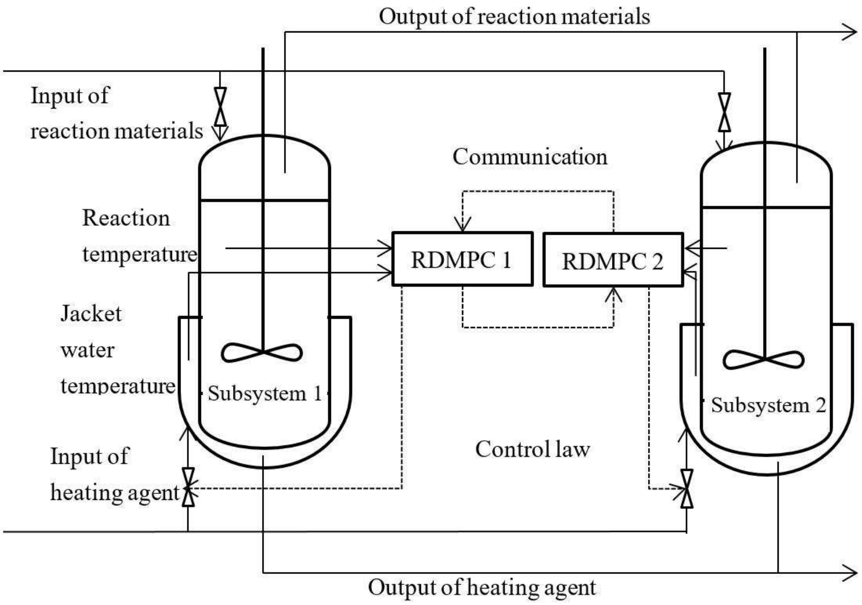

Figure 10: Simplified physical model of parallel CSTR system.

where:

$$a_{11} = -(\frac{\mathcal{U}A_R}{MC_{pC}} - \frac{(-\Delta H)M\bar{C}_A\bar{K}E}{V_R C_p R\bar{T}^2}) \quad a_{21} = \frac{\mathcal{U}A_R}{MC_p} \quad a_{12} = \frac{\mathcal{U}A_R}{V_C\rho_C C_{pC}}$$

$$a_{22} = -(\frac{\bar{W}}{V_C} + \frac{\mathcal{U}A_R}{V_C\rho_C C_{pC}}) \quad b_{22} = \frac{\bar{T}_H - \bar{T}_C}{V_C}$$

C_p is the specific heat of the reactants, $\bar{C}_A$ is the reactant concentration, M is the total mass of reactant, V_R is the volume of reactant, $\mathcal{U}$ is the total heat transfer coefficient, A_R is the heat transfer area, $\tilde{T}$ is the the temperature in reactor, ΔH is the Molar reaction heat, V_C is the volume of the heat agent, $\bar{T}_C$ is the outlet temperature of the heat agent, ρ_C is the density of the heat agent, C_{pC} is the specific heat of the heat agent, W is the flow rate of the heat agent, $\bar{T}_H$ is the inlet temperature of the heat agent, E is the activation energy, R is the molar gas constant, $\bar{K}$ is the reaction rate. The symbol "-" above the variables denotes a steady state value. The model is then given by:

$$\begin{aligned}
A_1 &= \begin{bmatrix} 0.9600 & 0.0039 \\ -0.2488 & 0.8902 \end{bmatrix}, \; A_2 &= \begin{bmatrix} 0.0722 & 0.0002 \\ -0.0134 & 0.0773 \end{bmatrix} \\
B_1 &= \begin{bmatrix} 0.0071 & 0.0001 \\ -0.0009 & 0.0285 \end{bmatrix}, \; B_2 &= \begin{bmatrix} 0.0097 & 0.0001 \\ -0.0004 & 0.0783 \end{bmatrix} \\
\sigma_{21} &= \begin{bmatrix} 0.0238 & 0.0412 \\ 0.1428 & 0.0981 \end{bmatrix}, \; \sigma_{12} &= \begin{bmatrix} 0.0351 & 0.0741 \\ 0.1024 & 0.0427 \end{bmatrix} \\
\rho_{21} &= \begin{bmatrix} 0.0416 & 0.0597 \\ 0.0736 & 0.1328 \end{bmatrix}, \; \rho_{12} &= \begin{bmatrix} 0.0842 & 0.0718 \\ 0.0981 & 0.0889 \end{bmatrix} \\
\delta_{21} &= \begin{bmatrix} 0.1243 & 0.2004 \end{bmatrix}, \; \delta_{12} &= \begin{bmatrix} 0.2181 & 0.1411 \end{bmatrix} \\
\lambda_{21} &= \begin{bmatrix} 0.1024 & 0.0971 \end{bmatrix}, \; \lambda_{12} &= \begin{bmatrix} 0.0518 & 0.0449 \end{bmatrix}
\end{aligned}$$

$$P_{11} = P_{21} = P_{12} = P_{22} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

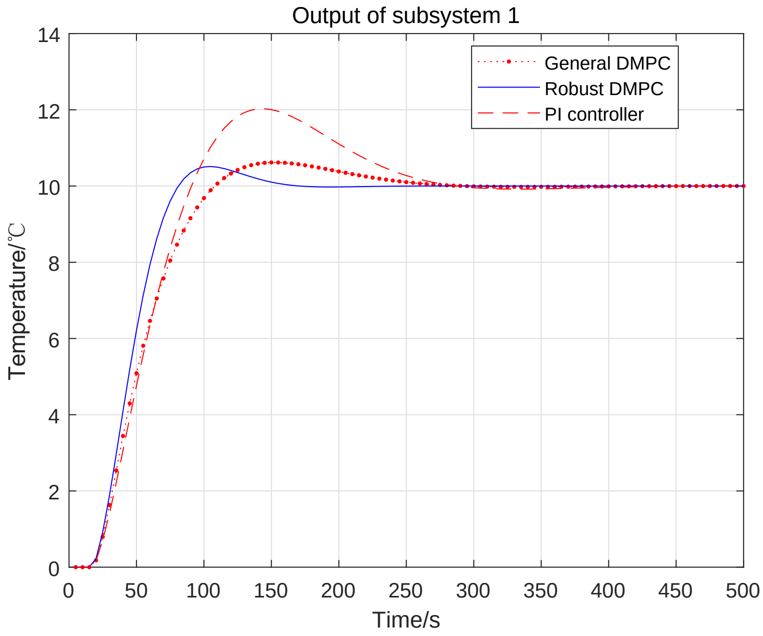

Figure 11: The output temperature of the first CSTR for the general DMPC, robust DMPC and PI implementations

Table 3: Computation time with a variable control horizon

M	computation time/s
1	2.95
2	3.96
3	5.38

,

$$R_{11} = R_{21} = R_{12} = R_{22} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

.

The sampling period $\mathbb{T}_s = 10s$ and the sampling length $\mathbb{N} = 200$. The predicted length $\mathbb{P} = 20$ and the control horizon $\mathbb{M} = 1$. $\alpha = 1$, $d_1 = 3$, $d_2 = 2$, $d_3 = 3$ and $d_4 = 2$. The PI parameters are set: P (K_c) is 0.8, I (T_i) is 0.5. The desired temperature is 10 degrees Celsius. The simulation results are shown from Fig. 11 to Fig. 14. To examine the impacts with a variable control horizon, there is a test conducted when control horizons are $\mathbb{M} = 1, 2, 3$, the results are shown from Fig. 16 to Fig. 17. Table 3 shows the computation time with a variable control horizon.

The simulation results show that the robust DMPC algorithm for parallel systems within process networks can effectively deal with problems with a rapid dynamic response. Fig. 11 and Fig. 12 show that the performance of the proposed approach is better than the general DMPC and PI control. Fig. 13 and Fig. 14 indicate that the general DMPC and PI control need larger control effort. The computing time for the robust DMPC is 2.95s while that of the general DMPC is 3.72s. Fig. 15 shows the robust DMPC control action for both subsystems. This clearly demonstrates how the competitive characteristics in parallel systems are accommodated by the proposed method: when u_1 is increasing, u_2 can be seen to correspondingly decrease to accommodate the competitive coupling and competitive constraints. Fig. 16 , Fig. 17 and Table 3 show that a larger control horizon will reduce control performance and increase the computation time.

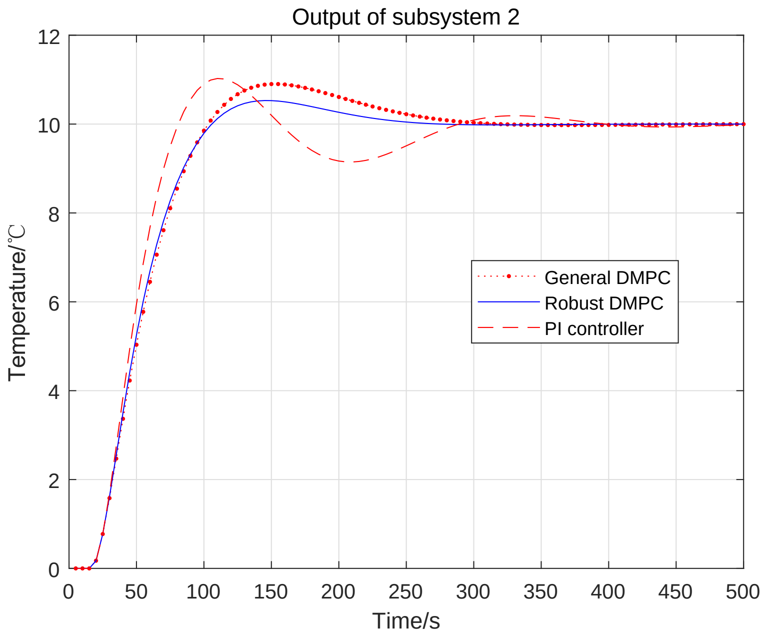

Figure 12: The output temperature of the second CSTR for the general DMPC, robust DMPC and PI implementations

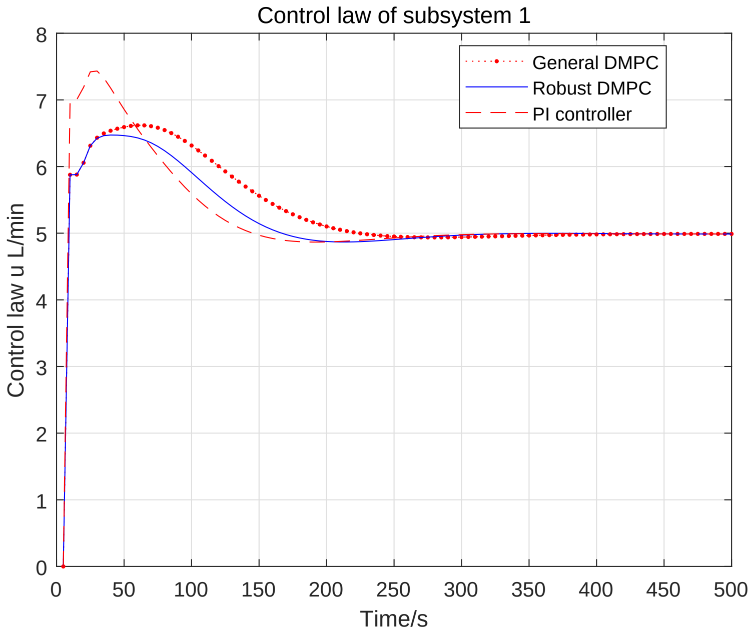

Figure 13: The control signals of the first CSTR for the general DMPC, robust DMPC and PI implementations

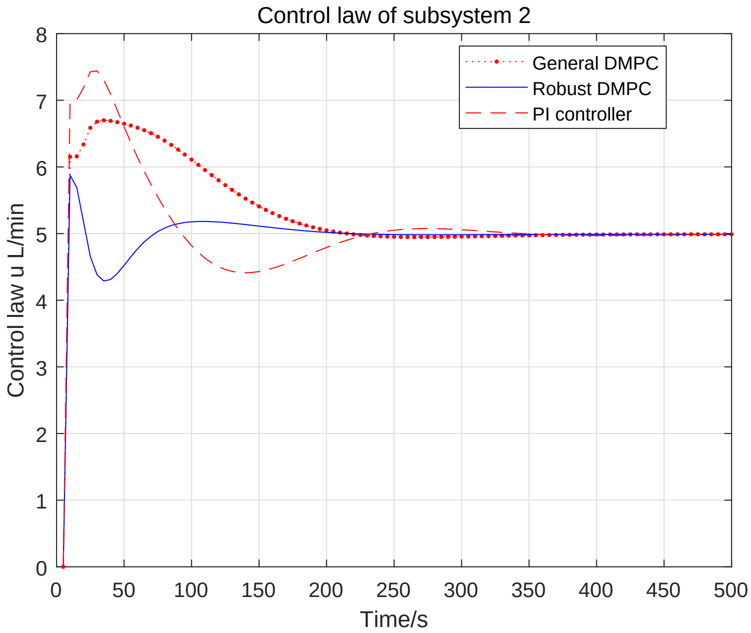

Figure 14: The control signals of the second CSTR system for the general DMPC, robust DMPC and PI implementations

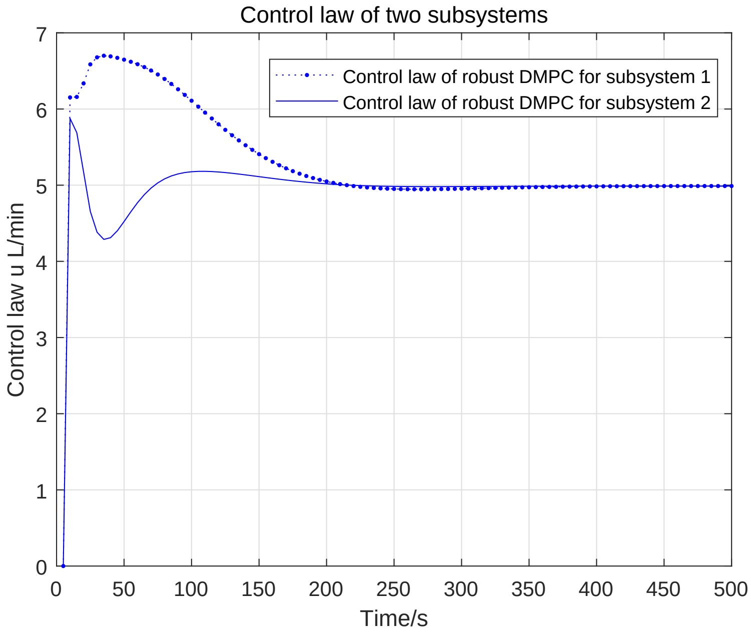

Figure 15: Control signal for both subsystems when controlled using the proposed robust DMPC

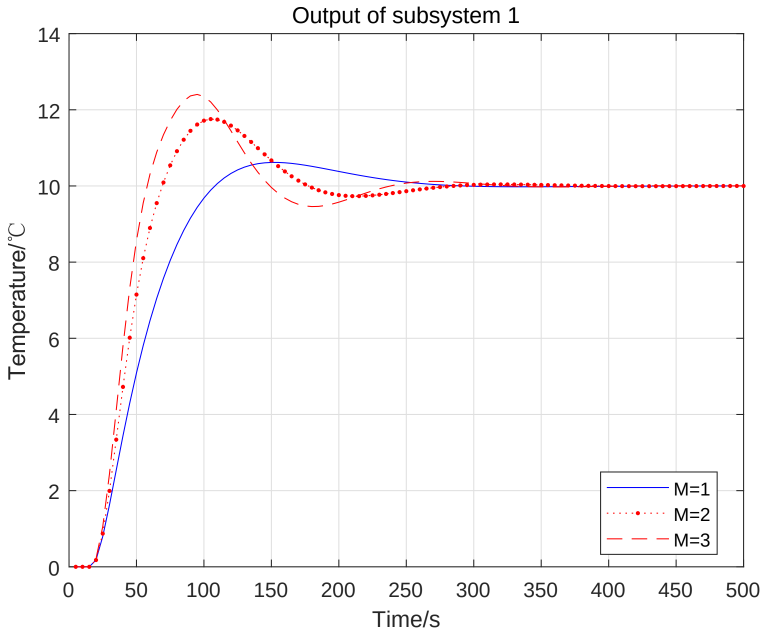

Figure 16: The output temperature of the first CSTR for the RDMPC implementations with a variable control horizon

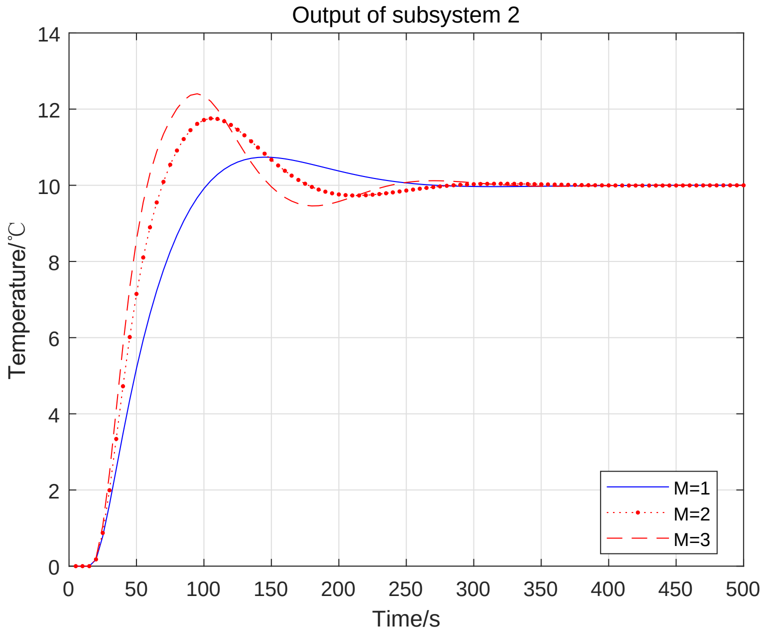

Figure 17: The output temperature of the second CSTR for the RDMPC implementations with a variable control horizon

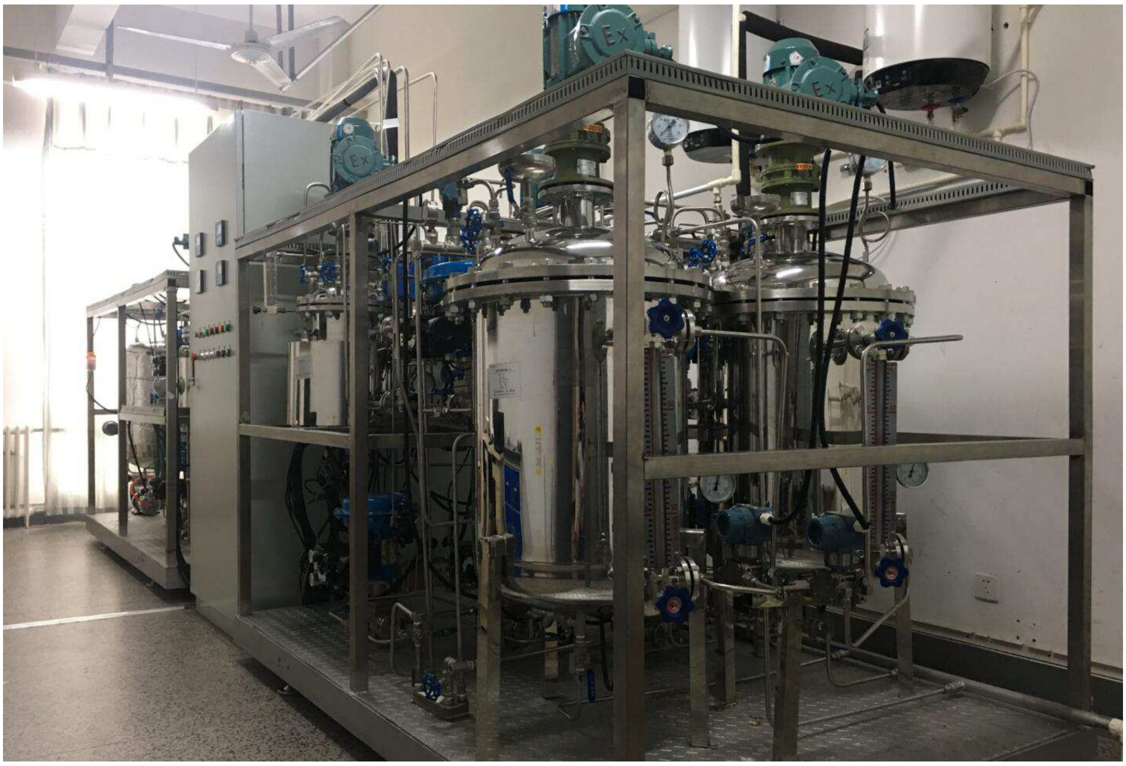

Figure 18: Experimental equipment

4.3. Parallel CSTR system experiment

The effectiveness of the robust DMPC algorithm for parallel CSTR systems has been verified by simulation. The effectiveness will be further validated by experiment. The Process Modelling and Control Group at the China University of Petroleum (East China) have developed an experimental rig which is shown in Fig. 18. The operation interface of the rig is shown in Fig. 19. The four reactors, labelled R101, R102, R103, R104, can be connected in numerous ways for controller validation and testing (series, parallel, series and parallel). The chemical reaction is carried out after feeding. The process can implement continuous operation as well as enable measurement and control of the flow, liquid level and temperature. V111 is the header tank which contains acetic ether and V112 is the header tank containing sodium hydroxide. These raw materials are processed in the CSTR at the same time. Unlike the previous simulation, this reaction is an exothermic reaction. Only two reactors are used in this experiment. The sampling period $\mathbb{T}_s = 10s$ and the sampling length $\mathbb{N} = 200$. The predicted length $\mathbb{P} = 20$ and the control horizon $\mathbb{M} = 1$, $\alpha = 1$, $d_1 = 3$, $d_2 = 2$, $d_3 = 3$ and $d_4 = 2$, $P_{11} = P_{21} = P_{12} = P_{22} = 1$, $R_{11} = R_{21} = R_{12} = R_{22} = 1$. For reactor R101, the desired set-point is 31 degrees Celsius and for reactor R102, the desired temperature is 33 degrees Celsius. The coolant flow in the jacket is used as the control variable. The initial temperatures of R101 and R102 are 28.1 degrees Celsius and 27.9 degrees Celsius, respectively. The robust DMPC algorithm is used for control of the system. The temperature tracking performance is shown in Fig. 20, which further validates the proposed approach.

For comparison purposes, the same experiment is conducted using a classical PID control scheme. The PID parameters are set according to the the ultimate-sensitivity method as in [41]: P (K_c) is 0.8, I (T_i) is 0.5, D (T_d) is 0.1. It can be seen from Fig. 22 that the controller struggles to accommodate the constraints and neither CSTR achieves the desired set-point. Comparing Fig. 20 with Fig. 21 it is seen that the robust DMPC algorithm exhibits significantly better performance than the PID method. The control input signals for the robust DMPC trial are shown in Fig. 22. There is much more reactant in R101 than in R102, so that R101 requires a greater control input to keep it stable. The variation of the control input required to reject the disturbance at time 10^4 can be clearly noticed. R102 contains less reactant and thus only only needs a small control input to reject the disturbance. In Fig. 20, the outputs

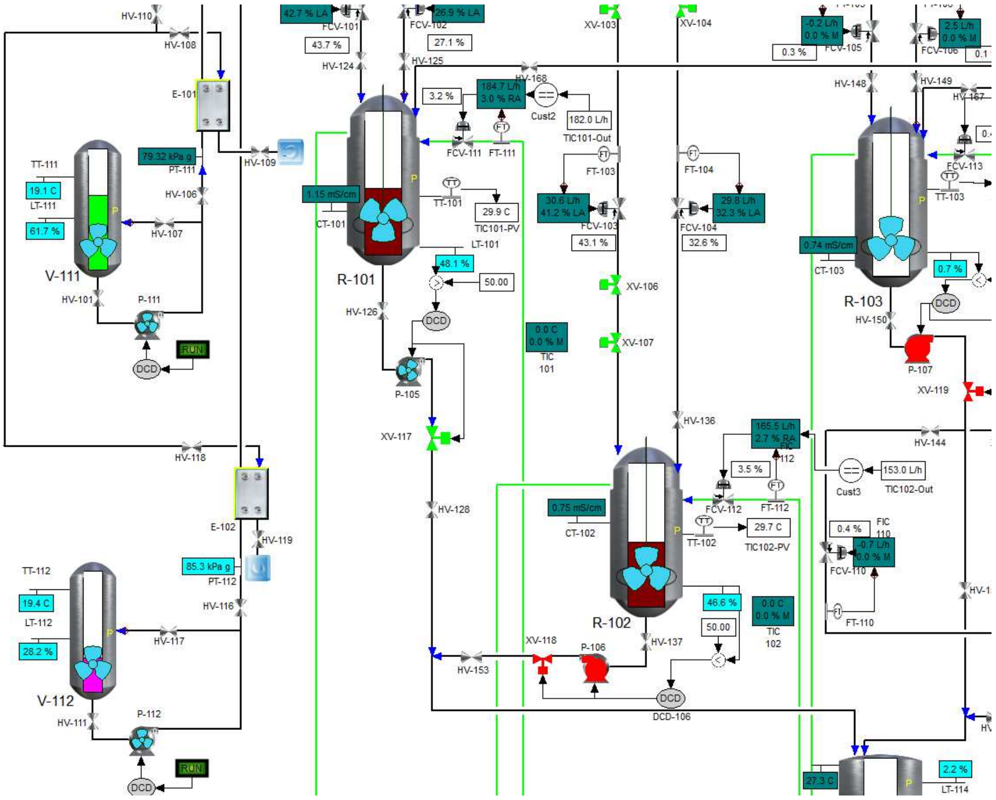

Figure 19: Parallel R101 and R102.

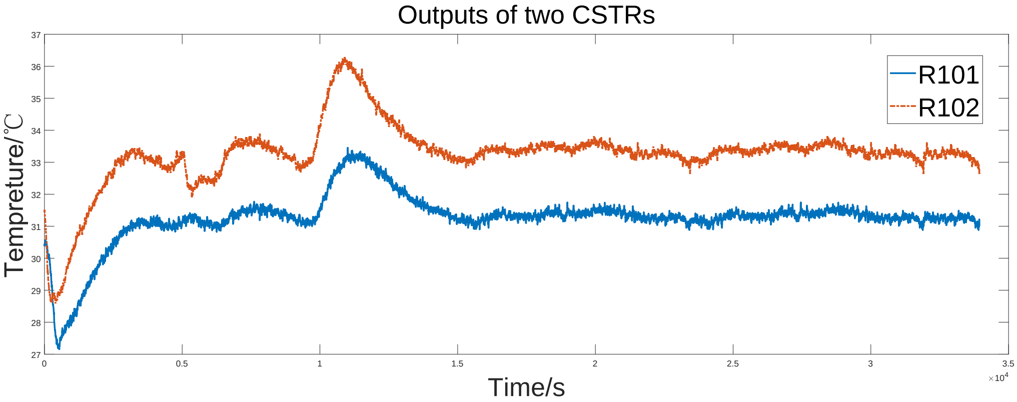

Figure 20: Tracking performance when the CSTR are controlled using the robust DMPC.

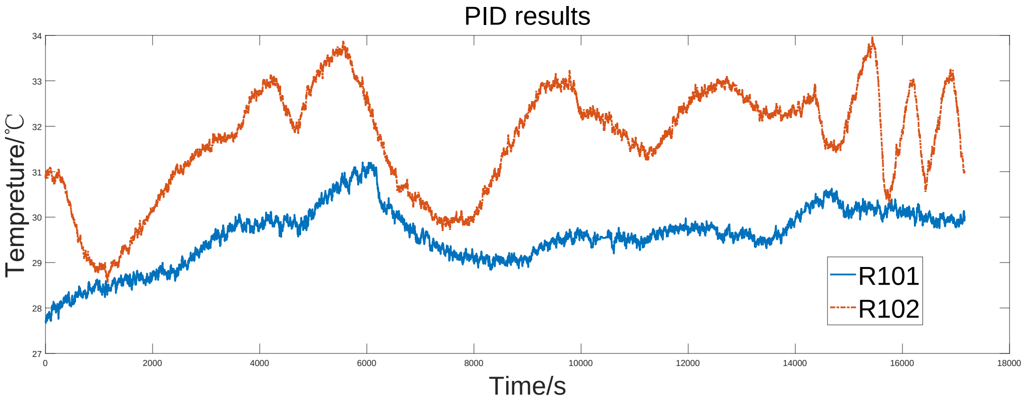

Figure 21: Tracking performance when the CSTR are controlled using a classical PID scheme.

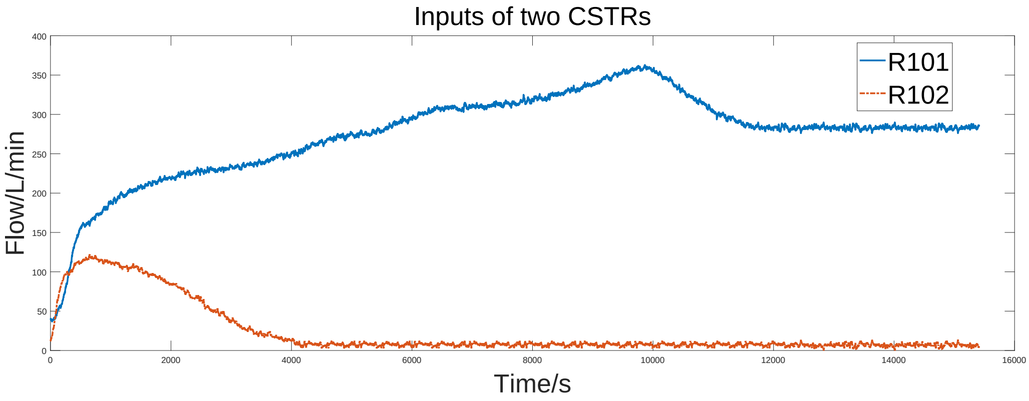

Figure 22: The two control input signals for the robust DMPC test.

have a significant variation at time $t = 10^4 s$ when an external disturbance is applied. The robust DMPC stabilises the outputs to the set-point under the disturbance. This shows the proposed approach has good robustness to external disturbances.

5. Conclusion

This paper has proposed an on-line algorithm to implement a robust DMPC strategy that explicitly accommodates the characteristics of parallel systems. The main contribution of this paper can be summarized as: (1) the competitive couplings and competitive constraints which describe the characteristics of parallel systems in process networks are defined; (2) a robust DMPC algorithm has been designed for parallel systems. The subsystem performance takes into account the couplings and constraints in order to achieve optimization of the whole system. The problem can be converted into N convex problems which can be expressed as linear matrix inequalities and solved iteratively by using the method of successive iteration to ensure fast convergence. It is also proven that if the algorithm is terminated at any feasible intermediate iteration, robust stability is still maintained. The simulations show that the proposed approach can achieve better performance than general DMPC and PI control for parallel systems. The results of an experimental trial further illustrate that the proposed approach is suitable for control of parallel systems in process networks.

Acknowledgement

This work is partially supported by the National Natural Science Foundation of China (61473312) and the Changjiang Scholar Project.

References

[1] P. Garcia-Herreros, A. Agarwal, J. M. Wassick, I. E. Grossmann, Optimizing inventory policies in process networks under uncertainty, Computers & Chemical Engineering 92 (2016) 256–272.

[2] D. B. Pourkargar, A. Almansoori, P. Daoutidis, Distributed model predictive control of process networks: impact of control architecture, IFAC-PapersOnLine 50 (1) (2017) 12452–12457.

[3] O. Weinstein, R. Semiat, D. Lewin, Modeling, simulation and control of liquid-liquid extraction columns, Chemical Engineering Science 53 (2) (1998) 325–339.

[4] S. Lucia, T. Finkler, S. Engell, Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty, Journal of Process Control 23 (9) (2013) 1306–1319.

[5] L. Chang, X. Liu, M. A. Henson, Nonlinear model predictive control of fed-batch fermentations using dynamic flux balance models, Journal of Process Control 42 (2016) 137–149.

[6] J. Širokỳ, F. Oldewurtel, J. Cigler, S. Prívara, Experimental analysis of model predictive control for an energy efficient building heating system, Applied energy 88 (9) (2011) 3079–3087.

[7] J. J. Downs, E. F. Vogel, A plant-wide industrial process control problem, Computers & chemical engineering 17 (3) (1993) 245–255.

[8] M. Morari, J. H. Lee, Model predictive control: past, present and future, Computers & Chemical Engineering 23 (4-5) (1999) 667–682.

[9] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. Scokaert, Constrained model predictive control: Stability and optimality, Automatica 36 (6) (2000) 789–814.

[10] C. E. Garcia, D. M. Prett, M. Morari, Model predictive control: theory and practiceła survey, Automatica 25 (3) (1989) 335–348.

[11] K. Fruzzetti, A. Palazoğlu, K. McDonald, Nolinear model predictive control using hammerstein models, Journal of process control 7 (1) (1997) 31–41.

[12] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, P. Haves, Model predictive control for the operation of building cooling systems, IEEE Transactions on control systems technology 20 (3) (2012) 796–803.

[13] S. L. de Oliveira Kothare, M. Morari, Contractive model predictive control for constrained nonlinear systems, IEEE Transactions on Automatic Control 45 (6) (2000) 1053–1071.

[14] Y. Wang, S. Boyd, Fast model predictive control using online optimization, IEEE Transactions on control systems technology 18 (2) (2010) 267.

[15] Q. Zhao, J. Jiang, Reliable state feedback control system design against actuator failures, Automatica 34 (10) (1998) 1267–1272.

[16] J. Han, From pid to active disturbance rejection control, IEEE transactions on Industrial Electronics 56 (3) (2009) 900–906.

[17] L. Liu, B. Huang, S. Dubljevic, Model predictive control of axial dispersion chemical reactor, Journal of Process Control 24 (11) (2014) 1671–1690.

[18] J. B. Rawlings, D. Q. Mayne, Model predictive control: Theory and design, Nob Hill Pub. Madison, Wisconsin, 2009.

[19] M. Morari, C. Garcia, J. Lee, D. Prett, Model predictive control, Prentice Hall Englewood Cliffs, NJ, 1993.

[20] J. M. Maciejowski, Predictive control: with constraints, Pearson education, 2002.

[21] A. Zakharov, E. Zattoni, M. Yu, S.-L. Jämsä-Jounela, A performance optimization algorithm for controller reconfiguration in fault tolerant distributed model predictive control, Journal of Process Control 34 (2015) 56–69.

[22] Y. Zheng, S. Li, H. Qiu, Networked coordination-based distributed model predictive control for large-scale system, IEEE Transactions on Control Systems Technology 21 (3) (2013) 991–998.

[23] A. Maxim, D. Copot, R. De Keyser, C. M. Ionescu, An industrially relevant formulation of a distributed model predictive control algorithm based on minimal process information, Journal of Process Control 68 (2018) 240–253.

[24] P. R. Mendes, J. M. Maestre, C. Bordons, J. E. Normey-Rico, A practical approach for hybrid distributed mpc, Journal of process Control 55 (2017) 30–41.

[25] M. Mercangöz, F. J. Doyle III, Distributed model predictive control of an experimental four-tank system, Journal of Process Control 17 (3) (2007) 297–308.

[26] C. Conte, C. N. Jones, M. Morari, M. N. Zeilinger, Distributed synthesis and stability of cooperative distributed model predictive control for linear systems, Automatica 69 (2016) 117–125.

[27] R. Scattolini, Architectures for distributed and hierarchical model predictive control–a review, Journal of process control 19 (5) (2009) 723–731.

[28] Y. Zhang, S. Li, Networked model predictive control based on neighbourhood optimization for serially connected large-scale processes, Journal of process control 17 (1) (2007) 37–50.

[29] H. Scheu, W. Marquardt, Sensitivity-based coordination in distributed model predictive control, Journal of Process Control 21 (5) (2011) 715–728.

[30] S. Zhang, D. Zhao, S. K. Spurgeon, X. Yan, Distributed model predictive control for the atmospheric and vacuum distillation towers in a petroleum refining process, in: 11th UKACC International Conference on Control, IEEE, 2016, pp. 1–6.

[31] P. D. Christofides, R. Scattolini, D. M. de la Pena, J. Liu, Distributed model predictive control: A tutorial review and future research directions, Computers & Chemical Engineering 51 (2013) 21–41.

[32] Q. Liu, H. S. Abbas, J. M. Velni, An lmi-based approach to distributed model predictive control design for spatially-interconnected systems, Automatica 95 (2018) 481–487.

[33] L. Zhang, J. Wang, Y. Ge, B. Wang, Robust distributed model predictive control for uncertain networked control systems, IET Control Theory & Applications 8 (17) (2014) 1843–1851.

[34] Z. Ge, Distributed predictive modeling framework for prediction and diagnosis of key performance index in plant-wide processes, Journal of Process Control 65 (2018) 107–117.

[35] A. Weinmann, Uncertain models and robust control, Springer Science & Business Media, 2012.

[36] Z. Wang, C. J. Ong, Distributed model predictive control of linear discrete-time systems with local and global constraints, Automatica 81 (2017) 184–195.

[37] M. V. Kothare, V. Balakrishnan, M. Morari, Robust constrained model predictive control using linear matrix inequalities, Automatica 32 (10) (1996) 1361–1379.

[38] W. Al-Gherwi, H. Budman, A. Elkamel, A robust distributed model predictive control algorithm, Journal of Process Control 21 (8) (2011) 1127–1137.

[39] L. Zhang, B. Huang, Robust model predictive control of singular systems, IEEE Transactions on Automatic Control 49 (6) (2004) 1000–1006.

[40] Z. Wan, M. V. Kothare, An efficient off-line formulation of robust model predictive control using linear matrix inequalities, Automatica 39 (5) (2003) 837–846.

[41] D. E. Rivera, M. Morari, S. Skogestad, Internal model control: Pid controller design, Industrial &amp; engineering chemistry process design and development 25 (1) (1986) 252–265.