Distributed model predictive control for constrained piecewise affine system with control coupling and its application in quadruple-tank

CONTROL PREDICTIVO DE MODELO DISTRIBUIDO PARA SISTEMAS RESTRINGIDOS DE ZONAS AFINES CON ACOPLAMIENTO DE CONTROL

DOI: http://dx.doi.org/10.6036/7740 | Recibido: 01/07/2015 • Aceptado: 01/09/2015

Guoqi Zhong[1,*], Zhiyuan Liu[1] and Houhua Jing[1,2]

[1] Department of Control Theory and Engineering, Harbin Institute of Technology. P. O. Box 3036, Science Park of Harbin Institute of Technology, No.2 Yikuang Street, Nangang District, 150080, Harbin, Heilongjiang, P.R. China, zhongguoqi@hit.edu.cn

[2] Department of Mechanical and Aerospace Engineering, College of engineering, The Ohio State University. 201 W 19th Ave W198 Columbus, OH 43210, U.S.A.

RESUMEN

Para mejorar el diseño de un controlador predictivo basado en un modelo para sistemas no lineales, se propone en este trabajo un modelo distribuido cooperativo (DMPC) por zonas afines (PWA) aplicado a un sistema compuesto por varios subsistemas restringidos. El esquema distribuido se aplica para ahorrar tiempo de cálculo de la solución. Para optimizar el rendimiento de todo el sistema, la cooperación tiene en cuenta la mayor parte de los objetivos de todo el sistema. El objetivo final, positivo e invariante, junto con el sistema en bucle cerrado se fijaron para garantizar la estabilidad. El problema óptimo de cada subsistema se transformó en una programación cuadrática mixta entera (MIQP) resuelto para obtener las acciones de control. El método propuesto descompone el problema de optimización de todo el sistema no lineal en varios MIQPs de menor dimensión de tal manera que el tiempo de cálculo se pueda reducirse drásticamente. Otra ventaja de la estrategia de optimización no lineal DMPC es que un problema no convexo podría ser sustituido por MIQPs finitos, pudiéndose obtener una aproximación al óptimo global en lugar de a un óptimo local. Finalmente se presenta un ejemplo numérico y la simulación de un sistema de cuatro tanques con el fin de poder evaluar el método propuesto.

- **Palabras clave:** Sistema por zonas afines, control distribuido, control predictivo basado en modelo, programación cuadrática mixta entera, sistema de cuatro tanques.

ABSTRACT

To improve the model predictive controller design of nonlinear system, a cooperative distributed model predictive control (DMPC) method for a system comprised of several constrained piecewise affine (PWA) subsystems with coupled control information is proposed in this paper. Distributed scheme is applied for saving time of solving optimization. To improve the optimal performance of entire system, cooperation was promoted with consideration to a greater portion of the system-wide objective. The terminal positively invariant set and cost of the entire closed-loop system were added to guarantee stability. The optimal problem of each subsystem was converted to mixed-integer quadratic programming (MIQP) solving for obtaining control actions. This method decomposed the optimization problem of a large-scale nonlinear system into several lower-dimension MIQPs such that solving time could be reduced dramatically. Another advantage of the DMPC strategy is non-convex nonlinear optimization could be replaced by finite MIQPs, and an approximate global optimum could be obtained rather than a local optimum. Finally, a numerical example and a quadruple-tank simulation are presented in order to assess the proposed DMPC strategy.

Keywords: Piecewise affine system, distributed control, model predictive control, mixed integer quadratic programming, quadruple-tank.

1. INTRODUCTION

Model predictive control (MPC) research began with the motivation of using computer technology to improve the control performance of systems that were multi-variable and had constraints [1]. In general, MPC appeared as a centralized control strategy. Nevertheless, for large-scale systems, a centralized model predictive controller that accounts for all interactions in the global system is either too complex or impractical. Typical examples of such systems are power networks, water networks, and urban traffic networks, among others. In order to overcome these shortcomings, decentralized and distributed control strategies have been developed in the MPC field [2].

DMPC is recognized as a highly practical control technology with high performance. It has been applied successfully to a water supply networks [3] and green-time control in urban traffic networks [4]. In a decentralized MPC, the original single optimization problem is replaced by a number of smaller ones, which have fewer decision variants and are easier to be solved in parallel. If the interactions among the subsystems are important or not small, the conditions that guarantee closed-loop nominal stability are strict [5]. In some cases, even the nominal closed-loop system behavior under a decentralized MPC may be unstable [6]. With the motivation of dealing with information exchanges between subsystems, a distributed MPC has captured scholars' attention. A communication-based DMPC scheme has been explored in the literature [7] and [8], where each subsystem's controller minimizes a local cost function for obtaining control actions; however, states sometimes move towards a Nash equilibrium with such a scheme. From a game theoretic perspective, the Nash equilibrium is a non-cooperative one, and is usually suboptimal in the Pareto sense [8]. In order to obtain improved optimized performance, a feasible cooperation-based model predictive control (FC-MPC) scheme has been developed in [9], [10], in which local objective functions of each subsystem have been modified to achieve system-wide control objectives. The FC-MPC strategy has two obvious advantages. The first is that the converged solution of this method is close to the Pareto optimal solution, from which the best achievable performance can be obtained through a centralized model predictive control (CMPC); the second advantage is that FC-MPC can deal with heavy interactions of state and control. In [11], Stewart proposes a stabilizability condition of the linear FC-MPC strategy, and also provides guidance on how to partition the subsystems within the plant. In [12], a novel distributed nonconvex optimization algorithm, the unique feature of which is that no coordinating optimization is required, is proposed for a nonlinear FC-MPC. This algorithm makes each iterate feasible and decreases the cost function. The convergence rate of the FC-MPC scheme has been investigated in [13], which reveals a strong connection between the weights, the strength of the coupling between subsystems, and the convergence rate. A singular value decomposition (SVD) framework for FC-MPC with highly coupled inputs was developed in [14], whose performance is close to that of the CMPC and outperforms the FC-MPC. The SVD structure can be decomposed into several independent subsystems and the coupling effect of the input and state variables can be overcome effectively.

Most investigations of DMPC concentrate on linear models. A huge obstacle when developing these schemes for nonlinear models is the fact that non-convex nonlinear optimization is hard to solve. Motivated by the above background, the work presented in this paper aims to contribute to the topic of distributed control of a coupled nonlinear system. Assume a plant comprised of several subsystems, and each subsystem is described by equation of:

$$\dot{x}_i = f_i(x) + \sum_{j=1}^{M} g_j(x)u_j. \tag{1}$$

It is well known that chemical plants, computer networks, and bio-systems are governed by very complex dynamics, so this class of system is common. Control is required to interconnect between subsystems. Due to the increasing requirements of control performance, such a non-linear model is essential for controller design. However, when a non-linear model is supplied for designing a DMPC controller, the optimization problem requiring a solution is a non-linear programming one; in most cases this means it is non-convex, time-consuming, and also difficult to find a global optimal solution. While linearization, a standard technique in the control community, is not always practical, an alternative approach is the PWA approximation, which has the ability to approximate non-linear functions with arbitrary precision. Given the above reasons, this work presents a DMPC scheme for constrained PWA subsystems. When approximating non-linear subsystems (1) with PWA models, the final optimization problem requiring a solution to obtain control actions is a mixed integer quadratic programming (MIQP) problem instead of a nonlinear programming problem. This change has two benefits. Firstly, MIQP is a convex optimization, requiring finite quadratic programming (QP) so an approximate global optimum can be obtained; on the other hand, distributed strategy can effectively reduce the number of decision variables of each MIQP, and decrease the time for solving optimization.

The paper is organized as follows. In Section 2, the DMPC problem is introduced. Next, the result of this paper is contained in Section 3. Section 4 reports the simulation results to corroborate our framework.

2. PROBLEM SETUP

In order to make this paper self-contained, some relevant definitions must be reviewed. $\mathbb{R}, \mathbb{Z}, \mathbb{Z}_+$ denote the field of real numbers, integers, and non-negative integers, respectively. Let $\mathbb{R}^n$ denote the vector space of the n-dimensions real numbers. Let $k \in \mathbb{Z}$ denote the discrete time index. Integers $N,\ M$ denote the predictive horizon and the number of subsystem, respectively. We used $\mathbb{I}_s$ to denote the finite integers set $\{1, \cdots, s\}$. In particular, $\mathbb{I}_{0:N}$ denotes the integer set $\{0, 1, 2, \cdots, N\}$. For given functions $\alpha_1(t)$ and $\alpha_2(t)$, $\alpha_1 \circ \alpha_2(t)$ denotes the function $\alpha_1(\alpha_2(t))$. For a set $\Omega \subseteq \mathbb{R}^n$, $cl(\Omega)$ denotes the closure of Ω.

In this paper, the plant we have considered is a class of a time-invariant discrete-time distributed system, and each subsystem has the PWA form, which can be obtained from the transformation of nonlinear sub-model (1) by suitable approximate method. To facilitate the exposition, we assumed the plant comprises only two subsystems, i.e. $M = 2$, and each system has multiple switching partitions. The analysis and result of more subsystems is similar. Consider the following system:

$$\begin{aligned} x_1^+ &= \begin{cases} A_1^1 x_1 + B_1^1 u_1 + W_{12}^1 u_2 + c_1^1 & x_1 \in \Omega_1^1 \\ \vdots & \vdots \\ A_1^{s_1} x_1 + B_1^{s_1} u_1 + W_{12}^{s_1} u_2 + c_1^{s_1} & x_1 \in \Omega_1^{s_1} \end{cases} \\ x_2^+ &= \begin{cases} A_2^1 x_2 + B_2^1 u_2 + W_{21}^1 u_1 + c_2^1 & x_2 \in \Omega_2^1 \\ \vdots & \vdots \\ A_2^{s_2} x_2 + B_2^{s_2} u_2 + W_{21}^{s_2} u_1 + c_2^{s_2} & x_2 \in \Omega_2^{s_2} \end{cases} \end{aligned} \tag{2}$$

For convenience, we have used x, u to denote the state and input at a given time instant and x^+ to denote the state at the next time instant. Indexes in superscript indicate the partition of the switching dynamic, and those in subscript indicate which subsystem they belong to. There is a little difference on the coefficient matrix W_{ij}^{s}. The subscript is divided into two parts. The first integer is the indicator of the current subsystem, and the second points out which subsystem the control u_j is derived from. Each partition is a polytope that has the following form:

$$\Omega_i^j = \left\{ x_i \in \mathbb{R}^{n_i} \mid G_{x_i}^j x_i \le K_{x_i}^j \right\},$$

where $i \in \mathbb{I}_2$, $j \in \mathbb{I}_{S_i}$. For the i-th subsystem, $\Omega_i^j \cap \Omega_i^r = \varnothing$, $\forall j, r \in \mathbb{I}_{s_i}, j \neq r$ and $\bigcup_{j\in\mathbb{I}_{s_i}} \Omega_i^j = \mathbb{X}_i$, which is state constraints of i-subsystem. $\mathbb{I}_{s_i}$ is the switching index set of i-th subsystem. When the origin of subsystem i is contained in partition Ω_i^j, then $c_i^j = 0$ is required to satisfy the continuity at point $x_i = 0$. The constraints of control are also polyhedral, namely:

$$u_i \in \mathbb{U}_i = \left\{ u_i \in \mathbb{R}^{m_i} \mid G_{u_i} u_i \le K_{u_i} \right\}, \ i \in \mathbb{I}_2.$$

The entire system is also PWA, but with more partitions, as the form of

$$x^+ = A^s x + B^s u + c^s \quad x \in \Omega^s \tag{3}$$

in which $A^s = \begin{bmatrix} A_1^j & 0 \\ 0 & A_2^r \end{bmatrix}, B^s = \begin{bmatrix} B_1^j & W_{12}^j \\ W_{21}^r & B_2^r \end{bmatrix}, c^s = \begin{bmatrix} c_1^j \\ c_2^r \end{bmatrix}$, $\Omega^s = \Omega_1^j \times \Omega_2^r$, and $s \in \mathbb{I}_S$, $S = S_1 S_2$. $\mathbb{I}_S$ is the switching index set of entire PWA system. The constraint set of the state and the inputs of the entire system are $\mathbb{X}$ and $\mathbb{U}$, respectively, which are also polytopes.

$$\mathbb{X} \triangleq \{x \in \mathbb{R}^n \mid G_x x \le K_x\} \tag{4}$$

$$\mathbb{U} \triangleq \{u \in \mathbb{R}^m \mid G_u u \le K_u\} \tag{5}$$

And $\mathbb{X} = \mathbb{X}_1 \times \mathbb{X}_2 = \left(\bigcup_{j\in\mathbb{I}_{s_1}} \Omega_1^j\right) \times \left(\bigcup_{r\in\mathbb{I}_{s_2}} \Omega_2^r\right), \mathbb{U} = \mathbb{U}_1 \times \mathbb{U}_2$. Moreover, we assume that the origin belongs to $\mathbb{X}$.

The goal of this paper is to design a DMPC strategy in order to stabilize the entire system with guaranteed bounds of states and controls of each subsystem.

3. DISTRIBUTED MODEL PREDICTIVE CONTROL

3.1. COST FUNCTION

Due to the existence of control information coupling among subsystems, a cooperative algorithm was applied to guarantee the stability of the entire closed-loop system and to obtain a better optimal performance. Consequently, in this paper, we used the global cost function, which has the following form:

$$J_N = \sum_{i=1}^{M} w_i \left(\sum_{k=0}^{N-1} L_i(x_i(k), u_i(k)) \right) + F(x(N)) \tag{6}$$

The stage function of i-th subsystem, namely $L_i(\cdot,\cdot)$, is usually defined using quadratic forms:

$$L_i(x_i(k), u_i(k)) \triangleq \|x_i(k)\|_{Q_i}^2 + \|u_i(k)\|_{R_i}^2 \tag{7}$$

as is the terminal cost:

$$F(x(N)) \triangleq \|x(N)\|_P^2 \tag{8}$$

where $Q_i \in \mathbb{R}^{n\times n}$, $i \in \mathbb{I}_M$ are positive semi-definite matrices and $R_i \in \mathbb{R}^{m\times m}$, $i \in \mathbb{I}_M$ are positive definite matrices.

As we have seen, the stage functions are defined for each subsystem, while the terminal cost is weighted on states of the entire system. The advantages for this are twofold. The most important advantage is to lower the conservative of linear matrix inequalities (LMI) which must be solved to obtain static control $h(x)$ and the terminal weighting matrix P; the other is convenient for adjusting the weighting coefficients of a subsystem's states. The stage function of entire system can be reformed as:

$$L\left(x(k), u(k)\right) = \|x(k)\|_Q^2 + \|u(k)\|_R^2 \tag{9}$$

in which $Q = \operatorname{diag}(w_1 Q_1, w_2 Q_2)$, $R = \operatorname{diag}(w_1 R_1, w_2 R_2)$.

3.2. STATIC CONTROL LAW, TERMINAL COST AND CONSTRAINT SET

In order to guarantee the stability of the entire closed-loop system, terminal positively invariant sets $\mathbb{X}_T$, weighting matrix P and static control law $u = h(x)$ must be designed according to entire system (3). Because of the switching characteristics of the PWA model, the design method of P and $h(x)$ that allow for arbitrary state switching from literature [15] was adopted.

Let $K \in \mathbb{R}^{m\times n}$ be a stabilizing static control gain for the entire system (3), then the static controller is:

$$h(x) = K^j x, x \in \mathbb{X}_T \cap \Omega^j, j \in \mathbb{I}_{S_0}, \tag{10}$$

and let $\mathbb{I}_{S_0} \triangleq \{j \in \mathbb{I}_S \mid 0 \in \operatorname{cl}(\Omega_j)\}$; that is, the index set of partitions that contain the origin. Because $h(x)$ is a centralized terminal static control law, let $h_i(x)$ denote the control law of i-th subsystem, namely, $u_i = h_i(x)$. Also, the entire system is piecewise linear in set $\cup_j \Omega^j$, $j \in \mathbb{I}_{S_0}$, then a closed-loop system in $\mathbb{X}_T$ with static controller (10) is:

$$x^+ = \left(A^j + B^j K^j\right) x \triangleq A_K^j x, \ x \in \mathbb{X}_T \cap \Omega^j, \ j \in \mathbb{I}_{S_0} \tag{11}$$

The terminal positively invariant set $\mathbb{X}_T$ of the entire system (3) subject to (10), not unexpectedly, $\mathbb{X}_T \subseteq \cup_{s\in\mathbb{I}_{S_0}} \Omega^s$. The terminal cost function is defined as follows:

$$F(x) = \begin{cases} x^T P^1 x & x \in \mathbb{X}_T \cap \Omega^1 \\ \vdots & \\ x^T P^s x & x \in \mathbb{X}_T \cap \Omega^s \end{cases} \tag{12}$$

A stability constraint of P^j and K^j can be solved by the LMIs as shown in [15]. The maximal positively invariant set $\mathbb{X}_T$ was calculated in the manner that iterated from 1 to infinity in [15]. When practically applied, a reasonable approximation of $\mathbb{X}_T$ is usually chosen.

3.3. CONVERSION OF OPTIMIZATION

With the static controller, terminal cost and constraint set described in the previous subsection, the distributed optimization problem of the i-th subsystem is given by following Problem $\mathcal{P}_i$,

$$\begin{aligned} &\mathbf{u}_i = \arg\min_{\mathbf{u}_i} J_i(x,\mathbf{u}) \\ &\text{s.t. all subsystem(2)} \\ &\quad x_i(0) = x_i, \quad i \in \mathbb{I}_M \\ &\quad (x_1(k),\cdots,x_M(k)) \in \mathbb{X},\ k \in \mathbb{I}_{0:N-1} \\ &\quad (u_1(k),\cdots,u_M(k)) \in \mathbb{U},\ k \in \mathbb{I}_{0:N-1} \\ &\quad x(N) \in \mathbb{X}_T \end{aligned} \tag{13}$$

It is inconvenient to predict future states by using the PWA model because of its switching characteristics. A mixed logic dynamic (MLD) model, which stems from [16] was introduced to construct state update equations for the subsystems. For the i-th subsystem, introduce logic variants δ_i^j and auxiliary variablez_i^j, $i \in \mathbb{I}_2, j \in \mathbb{I}_{S_i}$ as follows:

$$\left[x_i \in \Omega_i^j\right] \leftrightarrow \left[\delta_i^j = 1\right] \tag{14}$$

$$z_i^j = \left(A_i^j x_i + B_i^j u_i + W_{iq}^j u_q + c_i^j\right)\delta_i^j \tag{15}$$

then, the MLD models of the subsystems are:

$$x_i^+ = \sum_{j=1}^{S_i} z_i^j, \quad i \in \mathbb{I}_2 \tag{16}$$

Let $z_i \triangleq \left[(z_i^1)^T, (z_i^2)^T, \cdots, (z_i^{S_i})^T\right]^T$, then the state update equation of the i-th subsystem can be reformed as:

$$x_i^+ = \underbrace{\begin{bmatrix} I & I & \cdots & I \end{bmatrix}}_{S_i} z_i \tag{17}$$

Some constraints of δ_i^j, z_i^j are needed to guarantee equivalence between model transformations. This optimization problem (13) of subsystem $i \in \mathbb{I}_2$ can be written as a MIQP problem:

$$\begin{aligned} \min_{\mathbf{v}_i} J_i =& x(k)^T \mathcal{Y} x(k) + \mathbf{v}_i^T \mathcal{H}_i \mathbf{v}_i + \mathbf{v}_j^T \mathcal{S}_i \mathbf{v}_j + \mathbf{v}_i^T \mathcal{F}_i \mathbf{v}_j \\ \text{s.t.}\ \ & \mathcal{G}_i \mathbf{v}_i \le \mathcal{K}_{ix} x_i(0) + \mathcal{K}_{iu} \mathbf{u}_j + \mathcal{K}_{ic} \end{aligned} \tag{18}$$

where $j \in \mathbb{I}_2,\ j \neq i$. $\mathbf{v}_i(k) = \left[\mathbf{u}_i^T(k), \Delta_i^T(k), \mathbf{z}_i^T(k)\right]^T$ is the control of the MLD model of subsystem i, and $\Delta_i(k), \mathbf{z}_i(k)$ represent, respectively, auxiliary logical and continuous variables over predictive horizon, i.e.

$$\Delta_i(k) = \left[\delta_i(k)^T, \cdots, \delta_i(k+N-1)^T\right]^T$$
$$\mathbf{z}_i(k) = \left[z_i(k)^T, \cdots, z_i(k+N-1)^T\right]^T$$

The matrices $\mathcal{Y}, \mathcal{H}_i, \mathcal{S}_i, \mathcal{F}_i$ in the cost function and $\mathcal{G}_i, \mathcal{K}_{ix}, \mathcal{K}_{iu}, \mathcal{K}_{ic}$ in the constraint of the optimization problem (18) are matrices with suitable dimensions. The inequalities contain controls, states and terminal constraints and constraints that ensure equivalence between model transformations.

There are coupled variants both in the cost function and in the constraints of the optimization problem (18). When computing the optimization problem of the i-th subsystem at time k, because $u_j(k), j \neq i, j \in \mathbb{I}_2$ are unknown, it is replaced by $u_j(k|k-1)$; namely, the predicted u_j on the previous moment $k-1$.

Due to the existence of logical variables, the optimization problem (18) is a MIQP, and can be solved using existing optimization software, such as MOSEK, CPLEX, and MPT.

In order to ensure the recursive feasibility of the proposed DMPC, a warm start algorithm is a requisite. According to the optimum solution of problem (18) at time k, this algorithm is used to construct a feasible solution for the next moment.

Algorithm 1:

Given $x(k)$ and the optimal solution $\mathbf{v}_i^*(k)^T = \left[\mathbf{u}_i^*(k)^T, \Delta_i^*(k)^T, \mathbf{z}_i^*(k)^T\right]$, $i \in \mathbb{I}_M$, at time k, the warm start $\tilde{\mathbf{v}}_i(k+1)$ is calculated according to the rules below:

If $x(k) \in \mathbb{X}_T$,

then $[\hat{u}_1(k+j|k),\ \hat{u}_2(k+j|k), \cdots, \hat{u}_M(k+j|k)] = h(x(k+j|k)),\ j = \mathbb{I}_{0:N-1}$. $x(k+j|k)$ are computed according to Formula (3).

$$\begin{aligned} \tilde{\mathbf{u}}_i(k+1) &= [\hat{u}_i(k+1|k), \cdots, \hat{u}_i(k+N-1|k), \hat{u}_i(k+N|k)] \\ \tilde{\Delta}_i(k+1) &= [\delta_i(k+1|k), \cdots, \delta_i(k+N|k)] \\ \tilde{\mathbf{z}}_i(k+1) &= [z_i(k+1|k), \cdots, z_i(k+N|k)] \end{aligned}$$

where $\delta_i(k+1+j|k) = \tau_i(x_i(k+1+j|k)), z_i(k+1+j|k) = \eta_i(x_i(k+1+j|k), \hat{u}_i(k+1+j|k), \delta_i(k+1+j|k)), j = \mathbb{I}_{0:N-1}$. The function $\tau_i(\cdot)$ represents the process of determining δ_i according to the state x_i, as shown in Formula (14). Similarly, the function $\eta_i(\cdots)$ represents Formula (15).

else $[\hat{u}_1(k+N|k),\ \hat{u}_2(k+N|k), \cdots, \hat{u}_M(k+N|k)] = h(x(k+N|k))$,

$$\begin{aligned} \tilde{\mathbf{u}}_i(k+1) =& [u_i^*(k+1|k), \cdots, u_i^*(k+N-1|k), \hat{u}_i(k+N|k)] \\ \tilde{\Delta}_i(k+1) =& [\delta_i^*(k+1|k), \cdots, \delta_i^*(k+N-1|k), \tau_i(x(k+N|k))] \\ \tilde{\mathbf{z}}_i(k+1) =& [z_i^*(k+1|k), \cdots, z_i^*(k+N-1|k), \\ & \eta_i(x_i(k+N|k), \hat{u}_i(k+N|k), \delta_i(k+N|k))] \end{aligned}$$

end if

Based on the solution of this optimization problem (18) for each subsystem at time k, namely $\mathbf{v}_i^*(k)$, new control actions are constructed as

$$\bar{\mathbf{v}}_i(k) = w_i \mathbf{v}_i^*(k) + (1 - w_i)\tilde{\mathbf{v}}_i(k),$$

where $\sum_i^M w_i = 1,\ w_i > 0$. The control actions applied to the plant are:

$$u_i^*(k) = \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix} \bar{\mathbf{v}}_i(k), \quad i \in \mathbb{I}_M \tag{19}$$

Remark 1: According to the content of subsection 3.3, it is necessary to point out that the coupling of control information appearing in the PWA subsystem, after the model transformation, is implicit in auxiliary variables; if there exists state coupling, the method presented in this paper can also be applied.

From the above description, a procedure of applying the DMPC method can be divided into two stages, namely offline computation and online optimization. The complete algorithm can be schematized as follows.

Algorithm 2:

Step 1. Given the parameters $\mathbb{Q}_i$, $\mathbb{R}_i$, w_i, $i \in \mathbb{I}_M$, and then compute static control law K^j, terminal weighting matrices P^j and terminal positively invariant set $\mathbb{X}_T$.

Step 2. Compute coefficient matrices in MIQP (18).

Step 3. Given initial states $x_i(0)$, and choose an initial feasible $\mathbf{v}_i(0), i \in \mathbb{I}_M$, then set $k=0$.

Step 4. Compute $\mathbf{v}_i^*(k)$ by solving optimization problem (18) for all $i \in \mathbb{I}_M$.

Step 5. Compute control actions $u_i^*(k)$ according to Formula (19).

Step 6. Transmit $u_j^*(k)$ to each interconnected subsystem $j \neq i$, and then apply the control actions to all subsystems for obtaining the new states.

Step 7. Construct a feasible solution according to Algorithm 1. Set $k=k+1$ and go to **Step 4**.

In Algorithm 2, Step 1 to 2 belong to the offline stage, while the other steps belong to online stage.

3.4. ANALYSIS

In general, the analysis of the MPC method is necessary from two aspects: feasibility and stability. Feasibility of the optimization problem means there exists one input profile at each instant, not necessarily optimal, satisfying the constraints and such that the value of cost function is bounded. Stability means these control actions ensure the system works well. Due to the repeated solutions of the optimization problem given by (18), we need feasibility at each time $k \geq 0$. In the following, we provide a lemma on the feasibility of the optimization problem at each moment.

Lemma 1: The feasibility of the open-loop optimal control problem (18) at time $k=0$ implies its feasibility for all $k>0$.

Proof: For the entire PWA system, the control action $\mathbf{v}=(\mathbf{v}_1,\cdots,\mathbf{v}_M)$ is said to be feasible at time k with state x, if each $\mathbf{v}_i$, $i \in \mathbb{I}_M$ is a feasible solution for sub-optimization problem (18). The proof is a recursive procedure. First, assuming the sub-optimization problems (18) for $i \in \mathbb{I}_M$ are feasible at the moment k, then there exist M optimal sequences of controls at time k, denoted as $\mathbf{v}_i^*(k)=\left[\mathbf{u}_i^*(k)^T, \Delta_i^*(k)^T, \mathbf{z}_i^*(k)^T\right]^T$, $i \in \mathbb{I}_M$. Then, a set of candidate feasible solutions at time $k+1$ can be constructed according to Algorithm 1. $\hat{u}_i(k+j|k)$ satisfies the constraints because of the positively invariance of terminal set $\mathbb{X}_T$, as illustrated in [15]. As a result of that, $\tilde{\mathbf{u}}_i(k+1)$ satisfies various constraints in optimization problem (13). $\tilde{\Delta}_i(k+1)$ and $\tilde{\mathbf{z}}_i(k+1)$ are generated in accordance with (14) (15), respectively. Consequently, all constraints in optimization problem (18b) can be satisfied by $\tilde{\mathbf{v}}_i(k+1)$. Finally, $\tilde{\mathbf{v}}_i(k+1)$ is a feasible solution of optimization problem (18).

Besides Lemma 1, the initial feasibility of control action $\mathbf{v}_0$ is required to guarantee the feasibility of the optimization problem at all times. The initial feasibility always can be achieved by choosing $\mathbf{v}_0$.

Remark 2: According to the equivalence of the PWA and MLD systems illustrated in [17], optimization problem (13) and (18) are also equivalent.

The following theorem, based on the feasibility of the optimization problem, ensures the stability of the entire closed-loop system.

Theorem 1: Consider system (2) for a fixed $N \in \mathbb{Z}_+$, and suppose that the recursive feasibility of optimization problem (13) is satisfied; then the origin of the DMPC closed-loop system is Lyapunov asymptotically stable.

Proof: Consider cost function

$$V_N(x(k)) = J_N(x(k), \mathbf{u}^*(k)) \tag{20}$$

as a candidate Lyapunov function, where $\mathbf{u}^*=\{\mathbf{u}_1^*,\ \mathbf{u}_2^*,\cdots,\mathbf{u}_M^*\}$. To prove the asymptotic stability of the entire closed-loop system, we need to prove two aspects: attractivity and stability.

Attractivity: By solving LMI listed in [15], P^j and K^j, $j \in \mathcal{S}_0$ are obtained, which allow for arbitrary switching and according to this implies:

$$F(x(k+1)) - F(x(k)) \leq L(x(k), u(k))$$

established in $\mathbb{X}_T$. From the optimization we know that for all $i \in \mathbb{I}_M$

$$J_N(x(k),\tilde{\mathbf{u}}_1(k),\cdots,\mathbf{u}_i^*(k),\cdots,\tilde{\mathbf{u}}_M(k)) \leq J_N(x(k),\tilde{\mathbf{u}}_1(k),\cdots,\tilde{\mathbf{u}}_i(k),\cdots,\tilde{\mathbf{u}}_M(k))$$

can be obtained, then by convexity of function $J_N(\cdot,\cdot)$, we have

$$J_N(x(k),\mathbf{u}_1^*(k),\cdots,\mathbf{u}_i^*(k),\cdots,\mathbf{u}_M^*(k)) \leq J_N(x(k),\tilde{\mathbf{u}}_1(k),\cdots,\tilde{\mathbf{u}}_i(k),\cdots,\tilde{\mathbf{u}}_M(k))$$

The difference of $V_N(x(k))$ is

$$\begin{aligned} V_N(x(k+1)) - V_N(x(k)) =& J_N(x(k+1), \mathbf{u}^*(k+1)) - J_N(x(k), \mathbf{u}^*(k)) \\ \leq& J_N(x(k+1), \tilde{\mathbf{u}}(k+1)) - J_N(x(k), \mathbf{u}^*(k)) \\ \leq& -L(x(k), u(k)) \end{aligned}$$

Taking a limit to infinity on both terms, and by virtue of the positive definite nature of the cost function, it is derived that:

$$\lim_{k\to\infty} \|x(k)\| = 0, \quad \lim_{k\to\infty} \|u(k)\| = 0$$

Then, the system converges to the equilibrium point in $\mathbb{X}_T$.

Stability: Since $V_N(x(k))$ is a quadratic function with respect to $(x(k), \mathbf{v}(k))$, obviously, there exists a $\mathcal{K}$-functions α_1 such that:

$$\alpha_1(\|x(k)\|) \leq V_N(x(k))$$

Let k be an instant such that $x(k) \in \mathbb{X}_T$, and let j_k denote the index of switching at instant k. In the terminal set $\mathbb{X}_T$, the candidate Lyapunov function satisfies:

$$\begin{aligned} V_N(x(k)) \leq& \sum_{t=k}^{k+N-1} \left(\|x(t)\|_Q^2 + \|u(t)\|_R^2\right) + \|x(k+N)\|_P^2 \\ =& \sum_{t=k}^{k+N-1} x^T(t) \left(P^{j_k} - \left(A_K^{j_k}\right)^T P^{j_{k+1}} A_K^{j_k}\right) x(t) \\ =& x^T(k) P^{j_k} x(k) \end{aligned}$$

It follows $V_N(x(k)) \leq \alpha_2(\|x(0)\|)$, where $\alpha_2(\cdot)$ is a $\mathcal{K}$ -function. As $V_N(x(k+1)) - V_N(x(k)) \leq 0$, we have:

$$\alpha_1(\|x(k)\|) \leq V_N(x(k)) \leq V_N(x(0)) \leq \alpha_2(\|x(0)\|).$$

For $\forall \epsilon > 0$, define $\delta(\epsilon) = \alpha_2^{-1} \circ \alpha_1(\epsilon) > 0$, then $\|x(k)\| \leq \alpha_1^{-1} \circ \alpha_2(\|x(0)\|)$. If $\|x(0)\| \leq \delta(\epsilon)$, then $\|x(k)\| \leq \epsilon$. Depending on the Definition 4.1 in [18], the stability is proved.

4. ILLUSTRATIVE EXAMPLES

In order to verify the proposed method, a numerical simulation are presented in this section, whose equilibrium locates on a common boundary of multiple partitions; this example is used to illustrate the case in which state trajectories switch between partitions when states steer toward the equilibrium.

Consider a discrete-time state space model with two subsystems,

$$x_1^+ = \begin{cases} \begin{bmatrix} 1.20 & -0.6928 \\ 0.6928 & 0.40 \end{bmatrix} x_1 + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_2 & x_1 \in \Omega_1^1 \\ \begin{bmatrix} 0.40 & 0.6928 \\ -0.6928 & 0.40 \end{bmatrix} x_1 + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} u_2 & x_1 \in \Omega_1^2 \end{cases} \quad (21)$$

$$x_2^+ = \begin{cases} \begin{bmatrix} 0.6928 & -0.40 \\ 0.40 & 0.6928 \end{bmatrix} x_1 + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_2 & x_1 \in \Omega_2^1 \\ \begin{bmatrix} 0.6928 & 0.40 \\ -0.40 & 0.6928 \end{bmatrix} x_1 + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 1.2 \end{bmatrix} u_2 & x_1 \in \Omega_2^1 \end{cases} \quad (22)$$

The regions of the four switch partitions are polytopes, and their math descriptions are as follows:

$$\Omega_1^1 = \{x_1 \in \mathbb{R}^2 \mid [1,0]x_1 \geq 0,\ \|x_1\|_\infty \leq 3\}$$
$$\Omega_1^2 = \{x_1 \in \mathbb{R}^2 \mid [1,0]x_1 < 0,\ \|x_1\|_\infty \leq 3\}$$
$$\Omega_2^1 = \{x_2 \in \mathbb{R}^2 \mid [0,1]x_2 \geq 0,\ \|x_2\|_\infty \leq 3\}$$
$$\Omega_2^2 = \{x_2 \in \mathbb{R}^2 \mid [0,1]x_2 < 0,\ \|x_2\|_\infty \leq 3\}$$

The constraints of control are $\|u_i\| \leq 3,\ i = 1, 2$. The control objective is to transfer the states to the origin. The constrained optimal control problems are solved with weighting matrices $Q_1 = Q_2 = 10I_2, R_1 = R_2 = 1$ and initial states $x_0 = [-1, 1, 1, 1]^T$.

Because the origin is on the boundary of all switching partitions, the switching set is $\mathbb{I}_{S_0} = \{1, 2, 3, 4\}$. The terminal matrix and static controller are:

$$P = \begin{bmatrix} 1.4202 & -0.8963 & -0.0719 & -0.0270 \\ -0.8963 & 1.0184 & -0.0657 & 0.1318 \\ -0.0719 & -0.0657 & 0.4021 & 0.0545 \\ -0.0270 & 0.1318 & 0.0545 & 0.4034 \end{bmatrix}$$

$$K^1 = \begin{bmatrix} -2.7255 & -3.7006 & -2.7448 & 2.2082 \\ -2.7255 & -3.7006 & -2.7448 & 2.2082 \end{bmatrix}$$

$$K^2 = \begin{bmatrix} 40.5701 & 56.5231 & 40.7991 & -52.1564 \\ -45.8436 & -63.7054 & -44.3526 & 56.0250 \end{bmatrix}$$

$$K^3 = \begin{bmatrix} -13.5585 & -17.1813 & -4.9301 & 6.2869 \\ 8.1915 & 10.1248 & -3.8221 & 0.1203 \end{bmatrix}$$

$$K^4 = \begin{bmatrix} -29.0584 & -37.7373 & -24.1656 & 25.6765 \\ 39.6091 & 51.7631 & 35.2955 & -38.9834 \end{bmatrix}$$

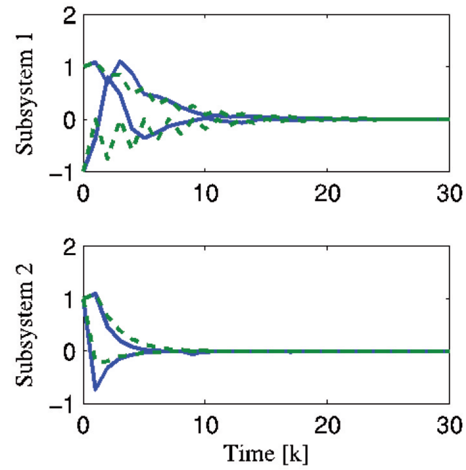

Fig. 1: Closed-loop state response of CMPC(green line) and DMPC (blue line).

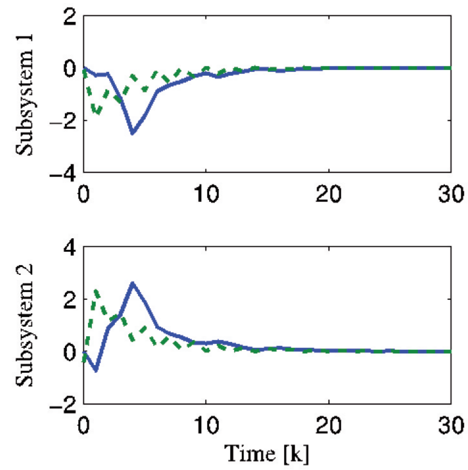

Fig. 2: The control trajectories of CMPC (green line) and DMPC (blue line).

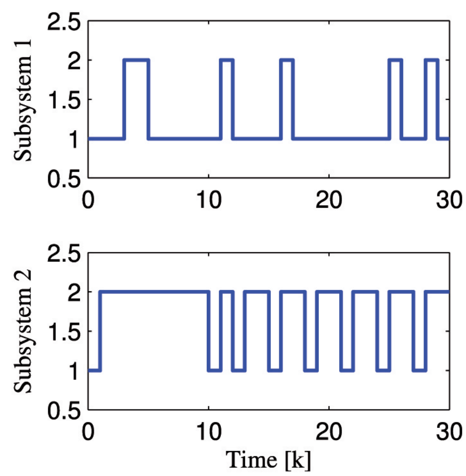

Fig. 3: The trajectories of logical variants of the DMPC.

The optimal closed-loop state and control trajectory for each subsystem with the proposed DMPC are depicted in Fig. 1 and 2, respectively. The solid lines are the simulation results with the DMPC strategy presented in the previous section, and the dashed lines are the result with a CMPC strategy as a comparison. The values of logical variants with DMPC are

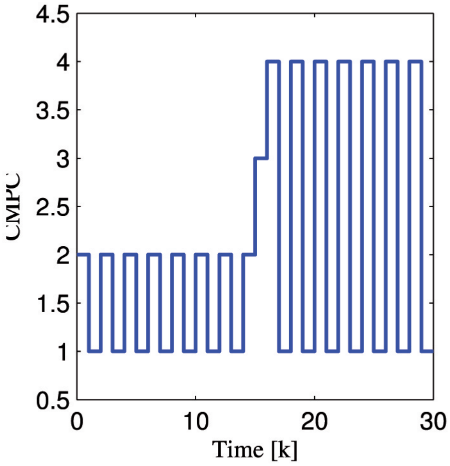

Fig. 4: The switching orders of the CMPC.

indicated in Fig. 3. For comparison, the switching sequence with a CMPC is shown in Fig. 4.

The DMPC controllers are successful in stabilizing all states of subsystems, and keeping the constrained states and controls within their limits. As can be seen from the trajectories of the states, after 15 steps the state gets close to the origin. However, because the origin is on the boundary of partitions, the states are always jumped between different partitions.

A CMPC simulation of an entire PWA system has been applied to compare the performance of the DMPC proposed in Section 3. Both the system and control curves of the CMPC twist more than that of the DMPC. Comparing Fig. 3 and 4, the reason for this is that the entire PWA system has a greater number of switching partitions and the behavior of transference of its states is consequently more complex.

When comparing calculation times, the DMPC has significant advantages. On a PC with Inter Core i5-3210M CPU (2.50 GHz) and 8.00 GB DDR3-1600 memory, the average solving times for 30 steps with different predictive times are listed in Table 1.

	DMPC	CMPC
$N=3$	0.0216	0.1007
$N=6$	0.0336	0.2997
$N=12$	0.0681	0.5967

Table 1: Comparing of Average Solving Time (Unit: s)

From Table I, with the increase of the predictive horizon, the total computation times show significant differences. The larger the predictive horizon is, the greater the difference is. In DMPC, the optimization problem of the entire system is decomposed into a number of smaller optimization problems, and each has fewer dimensions of decision variables. This leads to a significant decrease in the solving time.

5. APPLICATION TO QUADRUPLE-TANK SYSTEM

For deeply illustrating the detailed way to apply the DMPC proposed in the third section, it has been applied on a real quadruple-tank system in simulations. The complete design process of deploying the DMPC on a real nonlinear system is introduced. The quadruple-tank process is a typical minimum phase and a multi-variable system of interconnected tanks with nonlinear dynamics and is subject to state and input constraints [18]. This system can be used to verify complex advanced control and optimization algorithms, such as MPC, neural network control and fuzzy control. A real quadruple-tank system is shown in Fig. 5 and its scheme is presented in Fig. 6. The plant is composed of four tanks and two pumps. The two manipulated variables are the input voltages to the pumps, and the two controlled variables are the water levels of the lower two tanks.

The complete design process includes three steps. First, a nonlinear model of the controlled system is decomposed into several nonlinear distributed subsystems with suitable dimensions. Second, the subsystems are converted into PWA models by some approximate method and then are discretized. Finally, our proposed DMPC can be applied on these discrete-time PWA sub-models.

A nonlinear dynamic of quadruple-tank process can be modeled according to mass balance and Bernoulli's law.

$$\begin{aligned}
\dot{h}_1 &= -\frac{a_1}{A_1}\sqrt{2gh_1}+\frac{a_3}{A_1}\sqrt{2gh_3}+\frac{\gamma_1 k_1}{A_1}v_1\\
\dot{h}_2 &= -\frac{a_2}{A_2}\sqrt{2gh_2}+\frac{a_4}{A_2}\sqrt{2gh_4}+\frac{\gamma_2 k_2}{A_2}v_2\\
\dot{h}_3 &= -\frac{a_3}{A_3}\sqrt{2gh_3}+\frac{(1-\gamma_2)\,k_2}{A_3}v_2\\
\dot{h}_4 &= -\frac{a_4}{A_4}\sqrt{2gh_4}+\frac{(1-\gamma_1)\,k_1}{A_4}v_1
\end{aligned} \tag{23}$$

In Formula (23), $h_i,\ i\in\mathbb{I}_4$ are the levels of water in tank i considered as states, and $v_j,\ j=1,2$ are pump speeds, which are manipulated inputs. The physical meanings and values of other parameters appearing in (23) are described in the following Table II, from [18]. The state and control constraints considered in the quadruple-tank model (23) are $2\,cm \le h_i \le 30\,cm,\ i=1,2,3,4$ and $0\,V \le v_j \le 6\,V,\ j=1,2$, respectively.

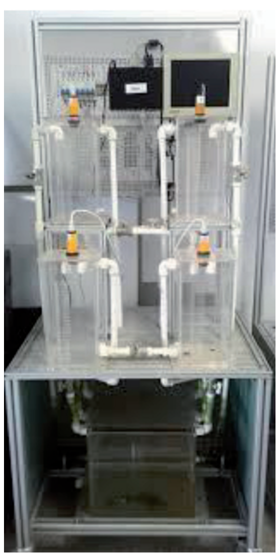

Fig. 5: The real quadruple-tank system.

	Value	Unit	Description
A_1	28	cm^2	The cross-section of tank 1.
A_2	32	cm^2	The cross-section of tank 2.
A_3	28	cm^2	The cross-section of tank 3.
A_4	32	cm^2	The cross-section of tank 4.
a_1	0.071	cm^2	The cross-section of the outlet hole of tank 1.
a_2	0.057	cm^2	The cross-section of the outlet hole of tank 2.
a_3	0.071	cm^2	The cross-section of the outlet hole of tank 3.
a_4	0.057	cm^2	The cross-section of the outlet hole of tank 4.
γ_1	0.70	-	The parameter of the three-way valve.
γ_2	0.60	-	The parameter of the three-way valve.
k_1	3.33	cm^3/Vs	The flow coefficient of the pump 1.
k_2	3.35	cm^3/Vs	The flow coefficient of the pump 2.
g	981	cm/s^2	The acceleration of gravity.

Table 2: The Parameters of Quadruple-Tank

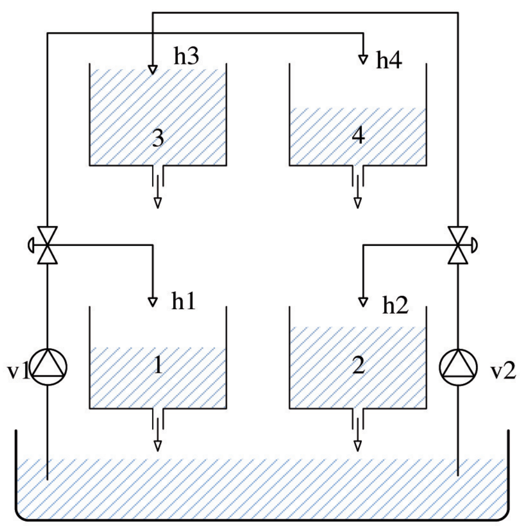

Fig. 6: The scheme of the quadruple-tank system.

The entire quadruple-tank nonlinear model (23) can be divided into two distributed subsystems. Subsystem 1 contains Tanks 1 and 3, and the other two tanks belong to subsystem 2, illustrated in Fig. 7. Namely:

$$\begin{bmatrix}\dot{h}_1\\ \dot{h}_3\end{bmatrix} = \begin{bmatrix}-\frac{a_1}{A_1} & \frac{a_3}{A_1}\\ 0 & -\frac{a_3}{A_3}\end{bmatrix}\begin{bmatrix}\sqrt{2gh_1}\\ \sqrt{2gh_3}\end{bmatrix} + \begin{bmatrix}\frac{\gamma_1 k_1}{A_1}\\ 0\end{bmatrix} v_1 + \begin{bmatrix}0\\ \frac{(1-\gamma_2)\,k_2}{A_3}\end{bmatrix} v_2 \qquad (24)$$

$$\begin{bmatrix}\dot{h}_2\\ \dot{h}_4\end{bmatrix} = \begin{bmatrix}-\frac{a_2}{A_2} & \frac{a_4}{A_2}\\ 0 & -\frac{a_4}{A_4}\end{bmatrix}\begin{bmatrix}\sqrt{2gh_2}\\ \sqrt{2gh_4}\end{bmatrix} + \begin{bmatrix}\frac{\gamma_2 k_2}{A_2}\\ 0\end{bmatrix} v_2 + \begin{bmatrix}0\\ \frac{(1-\gamma_1)\,k_1}{A_4}\end{bmatrix} v_1 \qquad (25)$$

Only control information interacts with each other in the distributed subsystems. Let $x_1 = (h_1, h_3)$ and $x_2 = (h_2, h_4)$ denote the states of subsystems, and let $u_1 = v_1$, $u_2 = v_2$ denote the control input.

Typically, the commonly used technique to design a controller for a nonlinear system is linearization at a given equilibrium. However, the approximate linear model is only accurate in the neighborhood of the equilibrium; that accuracy is poor anywhere else. In the case of a quadruple-tank, a PWA model is used instead of a linear model to approximate the nonlinear equation (23). The advantage is that an approximate PWA model has higher accuracy.

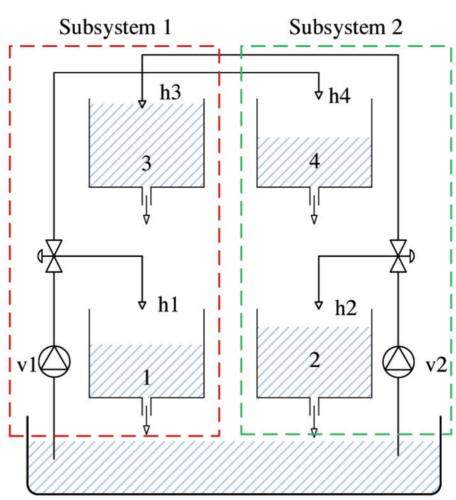

Fig. 7: The quadruple-tank process: two distributed subsystems

Notice that the nonlinearity of a quadruple-tank system (23) only reflects on $\sqrt{2gh_i}$, so if a PWA approximation of $\sqrt{2gh_i}$ can be calculated, we can obtain the PWA equation of the entire quadruple-tank dynamic. Using the method originating from literature [19], a PWA approximation is solved as follows:

$$\sqrt{2gh_i} \approx \begin{cases}\alpha_1 h_i + \beta_1 & h_{\min} \le h_i \le \gamma\\ \alpha_2 h_i + \beta_2 & \gamma < h_i \le h_{\max}\end{cases}. \qquad (26)$$

Parameters	Value
α_1	10.1850
α_2	5.0753
β_1	44.6396
β_2	94.3417
γ	9.7271
$h_{\min}$	1.00
$h_{\max}$	30.00

Table 3: The parameters of PWA approximation in (26)

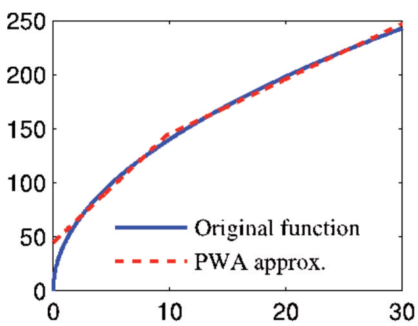

Fig. 8:The comparison between $\sqrt{2gh_i}$ and its PWA approximation.

where the parameters in (26) listed in Table 3. The reason only two region approximations were chosen in (26) is that fewer region approximations will reduce the complexity of the subsequent calculations, leading to fewer decision variants requiring solutions in the final MIQPs. Besides that, the approximation accuracy is also acceptable according to the comparison shown in Fig. 8. From that, the error between $\sqrt{2gh_i}$ and its PWA approximation is small on interval $[2,\ 30]$ which is the constraint considered in quadruple-tank process exactly.

Substituting the PWA approximation equation (26) into nonlinear equations (24-25) yields continuous-time PWA description for quadruple-tank subsystems. Using zero-order hold discretization, two discrete-time PWA sub-models of quadruple-tank can be obtained as follows:

The sample time we used is $5\,s$. Each subsystem has four switching partitions and the entire quadruple-tank model, which can be obtained via simple calculations according to foregoing method, has 16 partitions. The entire model is large and complicated and consequently not listed here.

The output equation of the entire system is:

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x \tag{29}$$

The aim in this example is to design a distributed predictive controller to stabilize the quadruple-tank system and regulate the output y to a given set point $\bar{y} = (25.4,\ 3.1)$. The steady state $\bar{x}$ and input $\bar{u}$ that tracks the set point can be determined using the solution of the quadratic programming listed in [20].

In this case, we need to design a tracking distributed predictive controller. The cost function we have adopted is:

$$\min_{\mathbf{u}_i}\ J_i(x, \mathbf{u}_1, \mathbf{u}_2) = \sum_{i=1}^{2} w_i \left(\sum_{k=0}^{N-1} \|x_i(k) - \bar{x}_i(k)\|^2_{Q_i} + \|u_i(k) - \bar{u}_i(k)\|^2_{R_i} \right) + \|x(k) - \bar{x}(k)\|^2_P \tag{30}$$

This cost function is different from (6). However, compared with the result in Section 3, only some minor changes are necessary to rebuild the MIQP. The initial state is $x_{10} = (12.4,\ 2.1), x_{20} = (11.7,\ 2.4)$. The setup parameters for the distributed predictive controller are: $Q_1 = Q_2 = 10 * I_2$, $R_1 = R_2 = 1$, $w_1 = w_2 = 0.5$. The prediction horizon has been taken as $N = 5$. The terminal matrix and static controller for the stability guarantee are listed below. Because the equilibrium is located in the interior of one region, only one static controller is needed.

$$P = \begin{bmatrix} 28.1856 & 6.6280 & -3.5986 & -26.3276 \\ 6.6280 & 63.1095 & -33.9352 & -21.5120 \\ -3.5986 & -33.9352 & 46.6058 & 13.7232 \\ -26.3276 & -21.5120 & 13.7232 & 94.8224 \end{bmatrix}; \tag{31}$$

$$x_1(k+1) = \begin{cases} \begin{bmatrix} 0.8789 & 0.1135 \\ 0 & 0.8789 \end{bmatrix} x_1(k) + \begin{bmatrix} 0.3905 \\ 0 \end{bmatrix} u_1(k) + \begin{bmatrix} 0.0142 \\ 0.2245 \end{bmatrix} u_2(k) + \begin{bmatrix} -0.0335 \\ -0.5309 \end{bmatrix} & x_1 \in \left\{ x \,\middle|\, \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} x \le \begin{bmatrix} 9.7271 \\ 9.7271 \\ -1 \\ -1 \end{bmatrix} \right\} \\ \begin{bmatrix} 0.8789 & 0.0584 \\ 0 & 0.9377 \end{bmatrix} x_1(k) + \begin{bmatrix} 0.3905 \\ 0 \end{bmatrix} u_1(k) + \begin{bmatrix} 0.0072 \\ 0.2317 \end{bmatrix} u_2(k) + \begin{bmatrix} 0.5551 \\ -1.1584 \end{bmatrix} & x_1 \in \left\{ x \,\middle|\, \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} x \le \begin{bmatrix} 9.7271 \\ 30 \\ -1 \\ -9.7271 \end{bmatrix} \right\} \\ \begin{bmatrix} 0.9377 & 0.1172 \\ 0 & 0.8789 \end{bmatrix} x_1(k) + \begin{bmatrix} 0.4031 \\ 0 \end{bmatrix} u_1(k) + \begin{bmatrix} 0.0145 \\ 0.2245 \end{bmatrix} u_2(k) + \begin{bmatrix} -0.6446 \\ -0.5309 \end{bmatrix} & x_1 \in \left\{ x \,\middle|\, \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} x \le \begin{bmatrix} 30 \\ 9.7271 \\ -9.7271 \\ -1 \end{bmatrix} \right\} \\ \begin{bmatrix} 0.9377 & 0.0603 \\ 0 & 0.9377 \end{bmatrix} x_1(k) + \begin{bmatrix} 0.4031 \\ 0 \end{bmatrix} u_1(k) + \begin{bmatrix} 0.0074 \\ 0.2317 \end{bmatrix} u_2(k) + \begin{bmatrix} -0.0369 \\ -1.1584 \end{bmatrix} & x_1 \in \left\{ x \,\middle|\, \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} x \le \begin{bmatrix} 30 \\ 30 \\ -9.7271 \\ -9.7271 \end{bmatrix} \right\} \end{cases} \tag{27}$$

$$x_2(k+1) = \begin{cases} \begin{bmatrix} 0.9133 & 0.0828 \\ 0 & 0.9133 \end{bmatrix} x_2(k) + \begin{bmatrix} 0.3002 \\ 0 \end{bmatrix} u_2(k) + \begin{bmatrix} 0.0067 \\ 0.1492 \end{bmatrix} u_1(k) + \begin{bmatrix} -0.0170 \\ -0.3801 \end{bmatrix} & x_2 \in \left\{ x \,\middle|\, \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} x \le \begin{bmatrix} 9.7271 \\ 9.7271 \\ -1 \\ -1 \end{bmatrix} \right\} \\ \begin{bmatrix} 0.9133 & 0.0422 \\ 0 & 0.9558 \end{bmatrix} x_2(k) + \begin{bmatrix} 0.3002 \\ 0 \end{bmatrix} u_2(k) + \begin{bmatrix} 0.0034 \\ 0.1526 \end{bmatrix} u_1(k) + \begin{bmatrix} 0.4050 \\ -0.8215 \end{bmatrix} & x_2 \in \left\{ x \,\middle|\, \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} x \le \begin{bmatrix} 9.7271 \\ 30 \\ -1 \\ -9.7271 \end{bmatrix} \right\} \\ \begin{bmatrix} 0.9558 & 0.0848 \\ 0 & 0.9133 \end{bmatrix} x_2(k) + \begin{bmatrix} 0.3071 \\ 0 \end{bmatrix} u_2(k) + \begin{bmatrix} 0.0068 \\ 0.1492 \end{bmatrix} u_1(k) + \begin{bmatrix} -0.4500 \\ -0.3801 \end{bmatrix} & x_2 \in \left\{ x \,\middle|\, \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} x \le \begin{bmatrix} 30 \\ 9.7271 \\ -9.7271 \\ -1 \end{bmatrix} \right\} \\ \begin{bmatrix} 0.9558 & 0.0432 \\ 0 & 0.9558 \end{bmatrix} x_2(k) + \begin{bmatrix} 0.3071 \\ 0 \end{bmatrix} u_2(k) + \begin{bmatrix} 0.0034 \\ 0.1526 \end{bmatrix} u_1(k) + \begin{bmatrix} -0.0184 \\ -0.8215 \end{bmatrix} & x_2 \in \left\{ x \,\middle|\, \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} x \le \begin{bmatrix} 30 \\ 30 \\ -9.7271 \\ -9.7271 \end{bmatrix} \right\} \end{cases} \tag{28}$$

$$K = \begin{bmatrix} -4.2301 \times 10^{-3} & -1.3321 \times 10^{-4} & -7.3409 \times 10^{-6} & -1.5362 \times 10^{-3} \\ 8.1954 \times 10^{-5} & -2.8135 \times 10^{-3} & -3.5537 \times 10^{-3} & -1.0706 \times 10^{-4} \end{bmatrix} \tag{32}$$

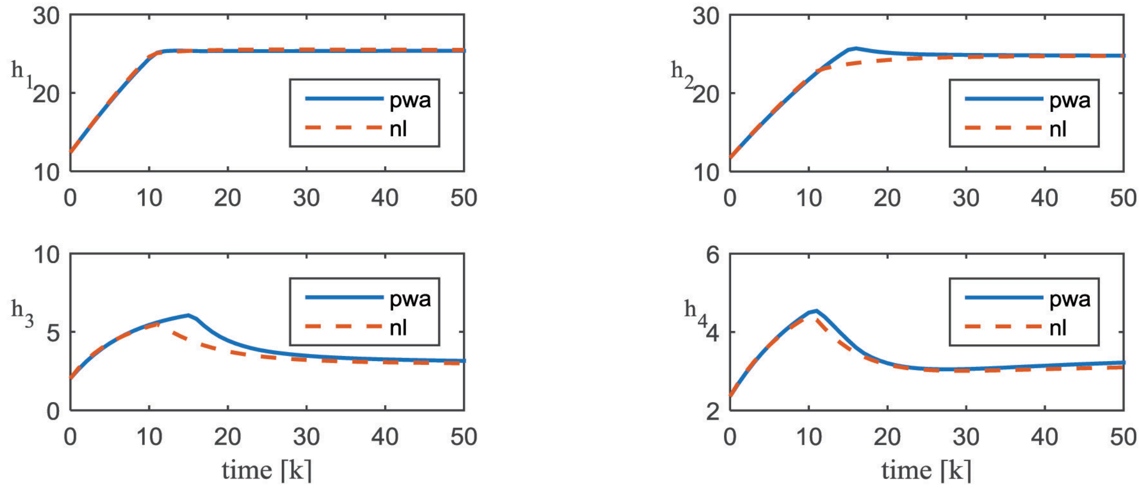

Fig. 9: The trajectories of states of the quadruple-tank.

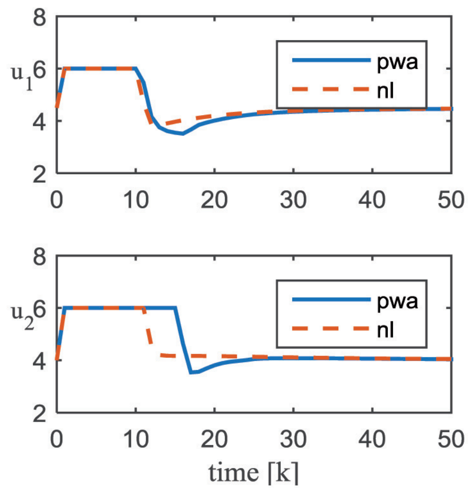

Fig. 10: The trajectories of controls of the quadruple-tank.

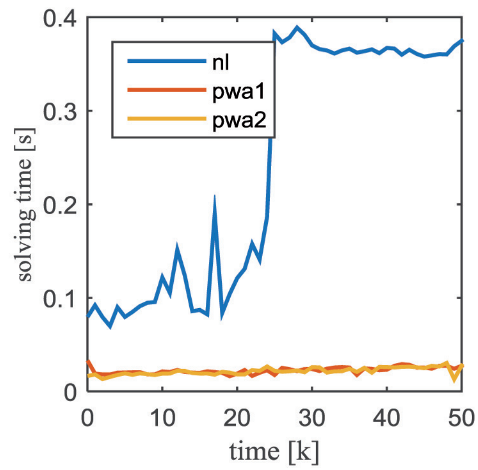

Fig. 12: The solving time comparison between nonlinear MPC and DMPC.

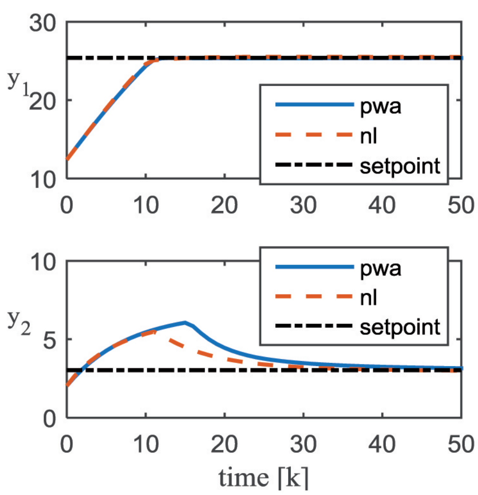

Fig. 11: The trajectories of outputs of the quadruple-tank.

For a deep understanding of the DMPC proposed in this paper, a comparison between the DMPC and a nonlinear model predictive control (NMPC) originated from Chen's work [21] was drawn simultaneously, which has been promoted as one of the most important contributions in the field. Chen presented a quasi-infinite horizon NMPC for nonlinear systems in [21], which guarantees asymptotic closed-loop stability. The simulation was run on MATLAB R2013a with an Inter Core i5-3210M CPU. The MIQP was solved using MOSEK 7.0. The NMPC in paper [21] needs to solve a nonlinear programming with nonlinear constraints, and fmincon function in optimization toolbox of MATLAB R2013a is suitable for carrying off it. The setup parameters of NMPC are the same with those used in the DMPC simulation.

The simulation results are presented in Fig. 9-12. In Fig. 11, the time evolution of the output is presented, where the desired set point is illustrated with black dot-dashed lines and the response curves of DMPC and NMPC are depicted in blue solid lines and orange dashed lines, respectively. In Fig. 9, the time evolution of h_1, h_2, h_3, h_4 is presented, and in Fig. 10, the time evolution of control input u_1 and u_2 is exhibited. In these figures, the blue solid lines and orange dashed lines denote the responses of DMPC and NMPC respectively. The solving time of these two methods is depicted in Fig. 12. The red and orange curves represent the time for solving the MIQPs of the two subsystems at each sample time, and the blue line indicates solving time of nonlinear programming. Notice that the curves of both states and controls fulfill the constraints we desired. From the comparison of the curves in

Fig. 9-12, the performance of NMPC is better than the performance of DMPC during the dynamic process, while in the steady-state process, their performance is very close. In addition, the DMPC has a huge advantage on time-consuming. The reason is that non-convex optimization and nonlinear constraints of NMPC greatly increase the solving time. These two difficult issues are overcame by the DMPC presented in this article as expected.

From this example, we can conclude that equilibrium of actual system is almost impossible to locate on a boundary of some partition. It means that only one partition contains the equilibrium in its interior. However, the equilibrium is considered as locating on the boundary, if it is very close to a boundary, in order to avoid oscillation of state trajectories.

5. CONCLUSION

In this work, a DMPC strategy has been presented for a system whose components consist of several nominal PWA models. To reduce the solving time of optimizations, a cooperative distributed framework was applied. For the convenience of the stability analysis, terminal inequality constraint set and terminal cost were employed. A warm start algorithm was used to achieve the recursive feasibility. The advantage of the DMPC strategy is that an approximation global optimum can be obtained, because the MIQP was solved through computing finite QPs. The controller has been applied to a numerical example and a quadruple-tank plant to prove the effectiveness of different situations. It should be noted that the stronger the non-linearity subsystem, the more partitions a PWA approximation needs, which leads to a more complex MIQP problem. The time for solving MIQP problems increased exponentially with the growth of the PWA partitions. In the practical application of the method proposed in this article, compromise between PWA approximation and optimized performance should be considered.

ACKNOWLEDGMENT

This study was funded by Project 61403105 which was supported by National Natural Science Foundation of China (NSFC).

BIBLIOGRAPHY

[1] Fan Liping, Zhang Jun, Li Chong. "Model predictive control on constant voltage output of a proton exchange membrane fuel cell". *Journal of Engineering Science and Technology Review*. April 2013. Vol. 6-2. p.115-119.
[2] Maestre JM, Muñoz De La Peña D, Camacho EF. "Distributed model predictive control based on a cooperative game". *Optimal Control Applications and Methods*. March 2011. Vol. 32-2. p.153-176. DOI: http://dx.doi.org/10.1002/oca.940
[3] Leirens S, Zamora C, Negenborn RR, et al. "Coordination in urban water supply networks using distributed model predictive control". *Proceedings of the 2010 American Control Conference*. June 2010. Vol. 2010. p.3957-3962. DOI: http://dx.doi.org/10.1109/ACC.2010.5530635
[4] De Souza F, Peccin VB, Camponogara E. "Distributed model predictive control applied to urban traffic networks: Implementation, experimentation, and analysis". *2010 IEEE Conference on Automation Science and Engineering (CASE)*. August 2010. Vol. 2010. p.399-405. DOI: http://dx.doi.org/10.1109/COASE.2010.5584213
[5] Christofides PD, Scattolini R, de la Pena DM, et al. "Distributed model predictive control: A tutorial review and future research directions". Computers & Chemical Engineering, April 2013. Vol. 51-5. p.21-41. DOI: http://dx.doi.org/10.1016/j.compchemeng.2012.05.011
[6] Rawlings JB, Stewart BT. "Coordinating multiple optimization-based controllers: New opportunities and challenges". Journal of process control. October 2008. Vol. 18-9. p.839-845. DOI: http://dx.doi.org/10.1016/j.jprocont.2008.06.005
[7] Camponogara E, Jia D, Krogh BH, et al. "Distributed model predictive control". IEEE *Control Systems Magazine*. February 2002. Vol. 22-1. p.44-52. DOI: http://dx.doi.org/10.1109/37.980246
[8] Jia D, Krogh BH. "Distributed model predictive control". *Proceedings of American Control Conference*. June 2001. Vol. 4. p.2767-2772. DOI: http://dx.doi.org/10.1109/ACC.2001.946306
[9] Venkat AN, Rawlings JB, Wright SJ. "Stability and optimality of distributed model predictive control". *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*. December 2005. Vol. 2005. p.6680-6685. DOI: http://dx.doi.org/10.1109/CDC.2005.1583235
[10] Venkat AN, Rawlings JB, Wright SJ. "Implementable distributed model predictive control with guaranteed performance properties". *Proceedings of the American Control Conference*. June 2006. Vol. 2006. p.613-618. DOI: http://dx.doi.org/10.1109/ACC.2006.1655424
[11] Stewart BT, Venkat AN, Rawlings JB, et al. "Cooperative distributed model predictive control". *Systems & Control Letters*. August 2010. Vol. 59-8. p.460-469. DOI: http://dx.doi.org/10.1016/j.sysconle.2010.06.005
[12] Stewart BT, Wright SJ, Rawlings JB. "Cooperative distributed model predictive control for nonlinear systems". *Journal of Process Control*. June 2011. Vol. 21-5. p.698-704. DOI: http://dx.doi.org/10.1016/j.jprocont.2010.11.004
[13] Groß D, Stursberg O. "On the convergence rate of a Jacobi algorithm for Cooperative Distributed MPC". *Proceedings of the IEEE Conference on Decision and Control*. December 2013. Vol. 2013. p.1508-1513. DOI: http://dx.doi.org/10.1109/CDC.2013.6760096
[14] Cai X, Sun P, Chen J, et al. "Rapid distributed model predictive control design using singular value decomposition for linear systems". *Journal of Process Control*. July 2014. Vol. 24-7. p.1135-1148. DOI: http://dx.doi.org/10.1016/j.jprocont.2014.05.002
[15] Lazar M, Heemels W, Weiland S, et al. "Stabilizing model predictive control of hybrid systems". *IEEE Transactions on Automatic Control*. Novermber 2006. Vol. 51-11. p.1813-1818. DOI: http://dx.doi.org/10.1109/TAC.2006.883059
[16] Bemporad A, Morari M. "Control of systems integrating logic, dynamics, and constraints". *Automatica*. March 1999. Vol. 35-3. p.407-427. DOI: http://dx.doi.org/10.1016/S0005-1098(98)00178-2
[17] Bemporad A. "Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form". *IEEE Transactions on Automatic Control*. May 2004. Vol. 49-5. p.832-838. DOI: http://dx.doi.org/10.1109/TAC.2004.828315
[18] Johansson KH. "The quadruple-tank process: A multivariable laboratory process with an adjustable zero". *IEEE Transactions on Control Systems Technology*. May 2000. Vol. 8-3. p.456-465. DOI: http://dx.doi.org/10.1109/87.845876
[19] Johansson M. *Piecewise linear control systems*. Springer, 2003. p.9-14. ISBN: 3-540-44124-7
[20] Muske KR, Rawlings JB. "Model predictive control with linear models". *AIChE Journal*, February 1993, Vol. 39-2. p.262-287. DOI: http://dx.doi.org/10.1002/aic.690390208
[21] Chen H, Allgöwer F. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*. October 1998. Vol. 34-10. p. 1205-1217. DOI: http://dx.doi.org/10.1016/S0005-1098(98)00073-9